1. **Create unique index on FacultyName on the Faculty collection.**

db.faculty.createIndex({facultyName:1})

1. **Using aggregation display the sum of final mark for all courses in Course collection.**

db.course.aggregate({$group:{\_id:"$courseName",course:{$sum:"$grade"}}})

**3.Implement relation between Student and Course, by adding array of Courses IDs in the student object.**

* + **Select specific student with his name, and then display his courses.**

**4.Implement relation between Student and faculty by adding the faculty object in the student using DBRef.**

* + **Select specific student with his name, and then display his faculty.**

1. **Display the count of students (use Group by with \_id: null, to not specify grouping column).**

db.student.aggregate({$group:{\_id:null, student:{$sum:1}}})

1. **Retrieve the total number of delivery days, grouped by year; retrieve the results only after 2017 (Hint: use aggregation pipelines)**

var pipe =[{$match:{year:{$gt:2017}}},{$group:{\_id:"$year",delivery:{$sum:"$delivery\_days"}}}]

db.o1.aggregate(pipe)

1. **Retrieve the total number of delivery days, grouped by**

**year; retrieve the results only paid**

var pipe =[{$match:{paid:"Y"}},{$group:{\_id:"$year",delivery:{$sum:"$delivery\_days"}}}]

db.o1.aggregate(pipe)

1. **Retrieve the total number of price, grouped by currency**

var pipe =[{$group:{\_id:"$cost.currency",price:{$sum:"$cost.price"}}}]

db.o1.aggregate(pipe)

1. **Calc how many record have color black**

var pipe =[{$match:{"items.colours":"black"}},{$group:{\_id:"$items.colours" ,color:{$sum:1}}}]

db.o1.aggregate(pipe)

{ \_id: [ [ 'blue', 'black' ] ], color: 3 }

{ \_id: [ [ 'pink', 'black' ], [ 'white' ] ], color: 1 }

{ \_id: [ [ 'white' ], [ 'white', 'black' ] ], color: 1 }

1. **Retrieve total all price from year 2017 to 2018**

var pipe = [

{$match:{$and :[{"$year":{$gte: 2017}}, {"$year":{$lte: 2018}}]}},

{$group: {\_id:null , price:{$sum: "$cost.price"}}}

]

db.o1.aggregate(pipe)

1. **How many product paid from 2018 to 2020?**

var pipe = [

{$match:{$and: [{"paid" : "Y"}, {"year": {$gte:2018}},{"year": {$lte:2020}}]}},{$group:{\_id:”$year” ,productpaid : {$sum:1}}}]

db.o1.aggregate(pipe)

1. **How many product currency nok and price greater than 20?**

var pipe= [

{$match : {$and : [{"cost.currency" : "NOK"},{"cost.price": {$gt:20}}]}},

{$group :{\_id : "$cost.currency", currencyNOK : {$sum:1}}}

]

db.o1.aggregate(pipe)

{

\_id: 'NOK',

currencyNOK: 2

}

**13. what is average delivery in 2020**

var pipe = [

{$match : {"year": 2020}},

{$group : {\_id :"$year",deliveryAvg : {$avg :"$delivery\_days" }}}

]

db.o1.aggregate(pipe)

**14. what is average price when delivery less than 4**

var pipe = [

{$match : {"delivery\_days": {$lt : 4}}},

{$group : {\_id :"$delivery\_days", priceAvg : {$avg :"$cost.price" }}}

]

db.o1.aggregate(pipe)