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

db.getCollection('faculty').createIndex({facultyName:1}, {unique: true})

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

db.getCollection('course').aggregate(

{$group:{\_id:'$courseName',finalMarks:{$sum:'$grade'}}}

)

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

db.getCollection('student').aggregate(

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

)

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

var pip = [

{$match:{year:{$gt: 2017}}},

{$group: {\_id: "$year", workingDays:{$sum: '$delivery\_days'}}}

]

db.orderesSection.aggregate(pip)

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

var pip = [

{$match:{paid:"Y"}},

{$group: {\_id: "$year", workingDays:{$sum: '$delivery\_days'}}}

]

db.orderesSection.aggregate(pip)

{ \_id: 2020, workingDays: 12 }

{ \_id: 2017, workingDays: 2 }

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

var pip = [

{$group: {\_id: "$cost.currency", workingDays:{$sum: '$cost.price'}}}

]

db.orderesSection.aggregate(pip)

{ \_id: 'NOK', workingDays: 66 }

{ \_id: 'EUR', workingDays: 46 }

1. **Calc how many record have color black**
2. **Retrieve total all price from year 2017 to 2018**