```
db.createCollection("salesman");
show collections
db.salesman.insert([{s_id:"5001",FirstName:"Thilak",LastName:"Perera",City:"Kandy",commision:
"0.15"}]);

db.salesman.insert([{s_id:"5002",FirstName:"Mohomed",LastName:"Abi",City:"Trinco",commision:
"0.18"},
{s_id:"5007",FirstName:"Dilruksha",LastName:"Wera",City:"Ampara",commision:"0.11"}]);

db.salesman.find({},{"FirstName":1});

db.salesman.updateMany({},{$unset:{"LastName":1}});

db.salesman.find({},{"FirstName":1,"_id":0})

db.salesman.find({},{"FirstName":1,"_id":0,"City":1})

db.salesman.updateMany({s_id:"5001"},{$set :{profit: 30}});

db.salesman.updateMany({s_id:"5009"},{$set :{profit: 29}});

db.customer.insertMany([{customerId: 1023, name: "Sagara"},{customerId: 1023, name:
"Nisansala"}]);

db.salesman.find({},{FirstName : 1, _id: 0}).limit(3)

db.salesman.find({},{FirstName : 1, _id: 0}).sort({FirstName: 1})

db.salesman.find({},{_id: 0}).sort({profit: -1})

db.salesman.find({},{ _id: 0}).sort({FirstName: 1, profit: -1})

db.salesman.find({$or :[{s_id:"5007"}, {FirstName:"Jagath"}]}, {_id: 0})

db.salesman.find({$and :[{s_id:"5009"}, {FirstName:"Jagath"}]}, {_id: 0})

db.salesman.find({$nor :[{FirstName:"Jagath"}, {City:"Oluvil"}]}, {_id: 0})

db.salesman.find({profit: {$lt:40}},{_id: 0})

db.salesman.find({$and: [{profit: {$lt:20}},{FirstName: "Dilruksha"}]},{_id: 0})

db.salesman.find({$and : [{profit: {$ne: 11}}, {FirstName: {$eq: "Thilak"}}]}, {_id: 0})

db.salesman.find({$and : [{profit: {$gte: 30}}, {profit: {$ne: 11}}]}, {_id: 0})

db.salesman.find({$and : [{profit: {$gte: 30}},{profit: {$ne: 11}}, {FirstName: {$eq: "Thilak"}}]}, {_id:
0})

db.salesman.find({FirstName: /^J/})

db.salesman.find().skip(2)

db.student.deleteOne({Name: "thanos"})
```

Lab 8_____

```
db.student.aggregate({$match: {section: "A"}})
```

```
db.student.aggregate([{$match: {section: "A"}}, {$group: {"_id": "$Section", totalcoursefee: {$sum: "$course_fee"}}}])

db.student.aggregate([{$match: {section: "A"}},{$count: "CountOfClassAstudents"}])



db.order.aggregate({$match: {size: "medium"}})

db.order.update({o_no: 3},{$set: {date: ISODate("2022-10-20")}},{$set: {quantity: 10}})

db.order.aggregate([{$match: {size: "medium"}},{$group: {"_id": "$type", totalquantity: {$sum: "$quantity"}}}])

 db.order.aggregate({$match: {date: {$lt: ISODate("2022-10-26"), $gt: ISODate("2022-10-15")}}})
```

Lab 09_____

```
db.student.aggregate({$match: {Section: "B"}},{$count: "TotalNoofSTUDENTS"})

db.student.aggregate([{$group: {"_id":"$Section", totalstudents: {$sum:1}, maximumage: {$max: "$Age"}, minimumage: {$min: "$Age"}}}])

db.student.find({Age: {$gt: 10}}, {_id: 0})

 db.student.aggregate({$sort: {"Age": 1}})

db.student.aggregate({$match: {Section: "B"}}, {$sort: {"Age": 1}}, {$skip: 2})

db.student.aggregate({$unwind: "$Subject"})
```

```
mongoimport Z:\books.json -d BOOKS -c books --drop
```

```
const { QuerySnapshot } = require("@google-cloud/firestore");
var admin = require("firebase-admin");

var serviceAccount = require("./serviceAccountKeys.json");

admin.initializeApp({
  credential: admin.credential.cert(serviceAccount)
});

const db = admin.firestore();
let empRef  = db.collection("employee");
empRef.get().then((QuerySnapshot) => {
    QuerySnapshot.forEach(document =>{
        console.log(document.data());
    })
}
)
const data = {
```

```
    age: 34,
    city: "abc",
    name: "ach"
    }
    db.collection("employee").doc(data.age.toString()).set(data);

    db.collection("employee").doc("0").delete().then( res => {
    console.log("document is deleted succesfully");
    })



npm install firebase-admin --save

node ./index.js

_____CREATE_____

const { query } = require("express");
const { database } = require("firebase-admin");
var admin = require("firebase-admin");

var serviceAccount = require("./serviceAccountKey.json");

admin.initializeApp({
  credential: admin.credential.cert(serviceAccount)
});

// const db=admin.firestore();
// let university = db.collection("seusl");

// university.get().then((query) =>{
//     query.forEach(document =>{
//         console.log(document.data());
//     })

// })
const db=admin.firestore();

let seusl=db.collection("seusl");

seusl.get().then((query) =>{
  query.forEach(document=>{
    console.log(document.data());
  })
}
)


_____INSERT_____

var admin = require("firebase-admin");

var serviceAccount = require("./serviceAccountKey.json");

admin.initializeApp({
  credential: admin.credential.cert(serviceAccount)
});
```

```
const db=admin.firestore();
const data = {
        id:3,
        name:"inu",
        faculty:"efac",
        department:"sl",
        yer:2023,
}
const data1={
        id:4,
        name:"gota",
        faculty:"fia",
        department:"sl",
        yer:2023,

}
const data2={
        id:5,
        name:"basil",
        faculty:"arts",
        department:"sl",
        yer:2023

}
//db.collection("seusl").doc(data.name.toString()).set(data);
db.collection("seusl").doc(data.id.toString()).set(data);
db.collection("seusl").doc(data1.id.toString()).set(data1);
db.collection("seusl").doc(data.id.toString()).set(data2)
```

_____DELETE_____

```
var admin = require("firebase-admin");

var serviceAccount = require("./serviceAccountKey.json");

admin.initializeApp({
  credential: admin.credential.cert(serviceAccount)
});

const db=admin.firestore();
db.collection("seusl").doc("ranil").delete().then(res=>{
    console.log("Delted ranil!")
})
```

_____UPDATE_____

```
var admin = require("firebase-admin");

var serviceAccount = require("./serviceAccountKey.json");

admin.initializeApp({
  credential: admin.credential.cert(serviceAccount)
});
```

```
const db=admin.firestore();

db.collection("seusl").doc("1").onSnapshot(update=>{
    console.log(update.data());
});
```