LAB-8

1. use Darshan

2. use DIET

3. show databases

4.db

5. db.dropDatabase()

6. db.createCollection("Student")  
7. db.createCollection("Department")  
8. show collections

9. db.Department.insertOne({Dname:'CE', HOD:'Patel'})  
10.db.Department.insertMany([{Dname:'IT'},{Dname:'ICT'}])

1

11.db.Department.drop()

12.db.Student.insertOne({Name:'Darshan',City:'Rajkot',Branch:'ce',Semester:4,Age:20})

13. db.Student.insertMany([

{Name:'Niyati',City:'Rajkot',Branch:'ce',Semester:4,Age:19},

{Name:'Darshan',City:'Rajkot',Branch:'me',Semester:6,Age:23},

{Name:'Yash',City:'Rajkot',Branch:'ce',Semester:3,Age:21}

])

14. db.getCollectionNames().includes("Student")

15. db.Student.stats()

16. db.Student.drop()

17. db.createCollection("Deposit")

18. db.Deposit.insertMany(

[

{ ACTNO: 101, CNAME: 'ANIL', BNAME: 'VRCE', AMOUNT: 1000.00, CITY: 'RAJKOT' },

{ ACTNO: 102, CNAME: 'SUNIL', BNAME: 'AJNI', AMOUNT: 5000.00, CITY: 'SURAT' },

{ ACTNO: 103, CNAME: 'MEHUL', BNAME: 'KAROLBAGH', AMOUNT: 3500.00, CITY: 'BARODA' },

{ ACTNO: 104, CNAME: 'MADHURI', BNAME: 'CHANDI', AMOUNT: 1200.00, CITY: 'AHMEDABAD' },

{ ACTNO: 105, CNAME: 'PRMOD', BNAME: 'M.G. ROAD', AMOUNT: 3000.00, CITY: 'SURAT' },

{ ACTNO: 106, CNAME: 'SANDIP', BNAME: 'ANDHERI', AMOUNT: 2000.00, CITY: 'RAJKOT' },

{ ACTNO: 107, CNAME: 'SHIVANI', BNAME: 'VIRAR', AMOUNT: 1000.00, CITY: 'SURAT' },

{ ACTNO: 108, CNAME: 'KRANTI', BNAME: 'NEHRU PLACE', AMOUNT: 5000.00, CITY: 'RAJKOT' }

]

)

19. db.Deposit.find()

20. db.Deposit.drop()

LAB-9

1. use BANK\_INFO

2. db.createCollection("Deposit")

3. db.Deposit.insertMany(

[

{

"ACTNO": 101,

"CNAME": "ANIL",

"BNAME": "VRCE",

"AMOUNT": 1000,

"ADATE": "1995-03-01"

},

{

"ACTNO": 102,

"CNAME": "SUNIL",

"BNAME": "AJNI",

"AMOUNT": 5000,

"ADATE": "1996-01-04"

},

{

"ACTNO": 103,

"CNAME": "MEHUL",

"BNAME": "KAROLBAGH",

"AMOUNT": 3500,

"ADATE": "1995-11-17"

},

{

"ACTNO": 104,

"CNAME": "MADHURI",

"BNAME": "CHANDI",

"AMOUNT": 1200,

"ADATE": "1995-12-17"

},

{

"ACTNO": 105,

"CNAME": "PRMOD",

"BNAME": "M.G. ROAD",

"AMOUNT": 3000,

"ADATE": "1996-03-27"

},

{

"ACTNO": 106,

"CNAME": "SANDIP",

"BNAME": "ANDHERI",

"AMOUNT": 2000,

"ADATE": "1996-03-31"

},

{

"ACTNO": 107,

"CNAME": "SHIVANI",

"BNAME": "VIRAR",

"AMOUNT": 1000,

"ADATE": "1995-09-05"

},

{

"ACTNO": 108,

"CNAME": "KRANTI",

"BNAME": "NEHRU PLACE",

"AMOUNT": 5000,

"ADATE": "1995-07-02"

}

]

)

1. db.Deposit.find()

2. db.Deposit.findOne()

3. db.Deposit.insertOne({

ACTNO: 109,

CNAME: 'KIRTI',

BNAME: 'VIRAR',

AMOUNT: 3000,

ADATE: '1997-03-05'

})

4. db.Deposit.insertMany([

{

ACTNO: 110,

CNAME: 'MITALI',

BNAME: 'ANDHERI',

AMOUNT: 4500,

ADATE: '1995-04-09'

},

{

ACTNO: 111,

CNAME: 'RAJIV',

BNAME: 'NEHRU PLACE',

AMOUNT: 7000,

ADATE: '1998-02-10'

}

])

5. db.Deposit.find({"BNAME":"‘VIRAR’"})

6. db.Deposit.find({"AMOUNT":{$gte:3000,$lte:5000}})

7. db.Deposit.find({$and:[{AMOUNT:{$gt:2000}},{BNAME:'VIRAR'}]})

8. db.Deposit.find({},{CNAME:1,BNAME:1,AMOUNT:1})

9. db.Deposit.find().sort({CNAME:1})

10. db.Deposit.find().sort({BNAME:-1})

11. db.Deposit.find().sort({ACTNO:1,AMOUNT:-1})

12. db.Deposit.find().limit(2)

13. db.Deposit.find().skip(2).limit(1)

14. db.Deposit.find().skip(5).limit(2)

15. db.Deposit.find().count()

Part-B

1. db.createCollection('Student')

db.Student.insertMany([{ \_id: 1, name: 'John', age: 30, city: 'New York', isActive: true },

{ \_id: 2, name: 'Jane', age: 25, city: 'Los Angeles', isActive: false },

{ \_id: 3, name: 'Tom', age: 35, city: 'Chicago', isActive: true },

{ \_id: 4, name: 'Lucy', age: 28, city: 'San Francisco', isActive: true },

{ \_id: 5, name: 'David', age: 40, city: 'Miami', isActive: false },

{ \_id: 6, name: 'Eva', age: 23, city: 'Boston', isActive: true },

{ \_id: 7, name: 'Nick', age: 38, city: 'Seattle', isActive: false },

{ \_id: 8, name: 'Sophia', age: 27, city: 'New York', isActive: true },

{ \_id: 9, name: 'Liam', age: 32, city: 'Los Angeles', isActive: false },

{ \_id: 10, name: 'Olivia', age: 29, city: 'San Diego', isActive: true }

])

2. db.Student.find()

3. db.Student.find({age:{$eq:30}})

4. db.Student.find({age:{$gt:30}})

5. db.Student.find({name:'John',age:{ $gt: 30 }})//error

6. db.Student.find({age:{$ne:25}})

7. db.Student.find({$or:[{age:25},{age:30},{age:35}]})

db.Student.find({age:{$in:[25,30,35]}})

8. db.Student.find({name:'John',age:{$eq:30}})

9. db.Student.find({name:'John',city:"New York"})

10. db.Student.find({name:'John',city:'New York'},{name:1,age:1})

**Part-c**

1. db.Student.find({age:{$gte:25,$lte:30}}).sort({age:1})

2. db.Student.find().sort({name:1,age:-1})

3. db.Student.find().limit(5)

4. db.Student.find().limit(2).skip(3)

5. db.Student.find().sort({age:-1}).limit(1)

6. db.Student.find().skip(2)

**Lab-10**

db.Student.insertMany([{ \_id: 1, name: 'John', age: 30, city: 'New York', isActive: true },

{ \_id: 2, name: 'Jane', age: 25, city: 'Los Angeles', isActive: false },

{ \_id: 3, name: 'Tom', age: 35, city: 'Chicago', isActive: true },

{ \_id: 4, name: 'Lucy', age: 28, city: 'San Francisco', isActive: true },

{ \_id: 5, name: 'David', age: 40, city: 'Miami', isActive: false },

{ \_id: 6, name: 'Eva', age: 23, city: 'Boston', isActive: true },

{ \_id: 7, name: 'Nick', age: 38, city: 'Seattle', isActive: false },

{ \_id: 8, name: 'Sophia', age: 27, city: 'New York', isActive: true },

{ \_id: 9, name: 'Liam', age: 32, city: 'Los Angeles', isActive: false },

{ \_id: 10, name: 'Olivia', age: 29, city: 'San Diego', isActive: true }

])

1.db.Student.updateMany({name:'John'},{$set:{age:31}})

2. db.Student.updateMany({city:'New York'},{$set:{city:'New Jersey'}})

3. db.Student.updateMany({age:{$gt:35}},{$set:{isActive:false}})

4. db.Student.updateMany({},{$inc:{age:1}})

5. db.Student.updateMany({name:'Eva'},{$set:{city:'Cambridge'}})

6. db.Student.updateMany({name:'Sophia'},{$set:{isActive:false}})

7. db.Student.updateMany({age:{$lt:30}},{$set:{city:'San Diego'}})

8. db.Student.updateMany({},{$rename:{age:'years'}})

9. db.Student.updateMany({name:'Nick'},{$set:{isActive:true}})

10. db.Student.updateMany({},{$rename:{country:'USA'}})

11. db.Student.updateMany({name:'David'},{$set:{city:'Orlando'}})

12. db.Student.updateMany({},{$mul:{years:2}})

13. db.Student.updateMany({name:'Tom'},{$unset:{city:''}})

14. db.Student.updateMany({years:{$gt:30}},{$set:{premiumUser:true}})

15. db.Student.updateMany({name:'Jane'},{$set:{isActive:true}})

16. db.Student.updateMany({name:'Lucy'},{$set:{isActive:false}})

17. db.Student.deleteOne({name:"Nick"})

18. db.Student.deleteMany({isActive:false})

19. db.Student.deleteMany({city:'New York'})

20. db.Student.deleteMany({years:{$gt:35}})

21. db.Student.deleteOne({name:"Olivia"})

22. db.Student.deleteMany({years:{$lt:25}})

23. db.Student.deleteOne({isActive:true})

24. db.Student.deleteMany({city:'Los Angeles'})

25. db.Student.deleteMany({city:{$exists:false}})

26. db.Student.updateMany({},{$rename:{"city":"location"}})

27. db.Student.updateMany({name:'John'},{$rename:{"name":"FullName"}})

28. db.Student.updateMany({},{$rename:{"isActive":"status"}})

29.db.Student.updateMany({city:'SanFrancisco'},{$rename:{"age":"yearsOld"})

30.db.creteCollectio(“Employee”,{Capped:true,size:5120,max:100,validator:{$jsonSchema:

{bsonType:”object”,

Required:[“Ecode”,”Ename”]

Properties:

Ecode:{bsonType:”int”}

Ename:{bsonType:”string”}

Age:{bsonType:”int”}

City:{bsonType:”string”}

}}}})

db.Employee.insertMany([

{"Ecode": 1, "Ename": "John"},

{"Ecode ": 2, "Ename": "Jane", "age": 25, "city": "Los Angeles"},

{"Ecode ": 3, "Ename": "Tom", "age": 35},

{"Ecode ": 4, "Ename": "Lucy", "age": 28, "city": "San Francisco", "isActive": true},

{"Ename": "Dino"}

])

**Part-B**

1. db.createCollection('Student\_data')

db.Student\_data.find({GENDER:'Female',CITY:'Rajkot'})

2. db.Student\_data.find({SEM:{$ne:3}})

3. db.Student\_data.find({CITY:{$in:['Jamnagar','Baroda']}})

4. db.Student\_data.find({CITY:'Baroda'},{SNAME:1}).limit(2)

5. db.Student\_data.find({GENDER:'Male',SEM:3})

6. db.Student\_data.find({ROLLNO:{$lt:105}},{SNAME:1,CITY:1,FEES:1})

7.db.Student\_data.updateMany({CITY:'Jamnagar',DEPARTMENT:'CE'},{$set:{CITY:'Surat'}})

8.db.Student\_data.updateMany({GENDRE:{$not:{$eq:'Femal'}}},{$inc:{FEES:500}})

9.db.Student\_data.updateMany({DEPARTMENT:'EE',SEM:3},{$set:{DEPARTMENT:'Electrical'}}

10. db.students.updateMany( { city: "Rajkot", // Filter for students in 'Rajkot' gender: "Male" // Filter for male students }, { $set: { fees: NEW\_FEE\_AMOUNT } // Replace 'NEW\_FEE\_AMOUNT' with the desired fee value } );

11. db.Student\_data.updateMany({SEM:5,FEES:{$lt:10000}},{$set:{CITY:'Vadodara'}})

12. db.Student\_data.deleteMany({$or:[{CITY:'Ahmedabad'},{GENDRE:'Male'}]})

13. db.Student\_data.deleteMany({ROLLNO:{$nin:[101,105,110]}})

14. db.Student\_data.deleteMany({$and:[{SEM:{$nin:[5,7]}},{DEPARTMENT:'Civil'}]})

15. db.Student\_data.deleteMany({CITY:{$nin:['Rajkot','Baroda','Jamnagar']}})

16. db.Student\_data.deleteMany({ROLLNO:{$gte:105,$lte:108}})

17. db.Student\_data.updateOne({},{$rename:{'CITY':'LOCATION'}})

18. db.Student\_data.updateOne({FEES:{$lte:10000}},{$rename:{'DEPARTMENT':'Branch'}})

19. db.Student\_data.updateMany({ROLLNO:{$in:[106,107,108]}},{$rename:{'SNAME':'Fullname'}})

20. db.Student\_data.updateOne({FEES:{$gte:9000}},{$rename:{'FEES':'Tuition\_Fees'}})

21. db.students.updateMany({ fees: { $lt: 15000 }, GENDRE: 'Female' },{ $rename: { "Department": "Major" } } );

22.db.Student\_data.updateMany({SEM:3,Major:{$ne:'Mechanical'}},{$rename:{'LOCATION':'Hometown'}})

**LAB-11**

**PART-A**

1. db.Employee.find({ENAME:/^E/})

2. db.Employee.find({ENAME:/n$/})

3. db.Employee.find({ENAME:/^[SM]/})

4. db.Employee.find({CITY:/^[A-M]/})

5. db.Employee.find({CITY:/ney$/})

6. db.Employee.find({ENAME:/n/i})

7. db.Employee.find({ENAME:/^E.{4}$/})

8. db.Employee.find({ENAME:/^S.\*a$/})

9. db.Employee.find({ENAME:/^phi/},{EID:1,ENAME:1,CITY:1,SALARY:1})

10. db.Employee.find({CITY:/dne/},{ENAME:1,JOININGDATE:1,CITY:1})

11.db.Employee.find({CITY:{$not:{$in:['London','Sydney']}}},{ENAME:1,JOININGDATE:1,CITY:1})

12. db.Employee.find({ENAME:/^J/})

13. db.Employee.find({ENAME:/y$/})

14. db.Employee.find({ENAME:/a/})

15. db.Employee.find({ENAME:/[ae]/})

16. db.Employee.find({ENAME:/^J.\*n$/})

17. db.Employee.find({CITY:/^New/})

18. db.Student.find({CITY:{$not:/^L/}})

19. db.Employee.find({CITY:/York/})

20. db.Employee.find({ENAME:/[aeiou]{2}/})

21. db.Employee.find({ENAME:/^.{3,}$/})

22. db.Employee.find({ENAME:/^.{4}$/})

23. db.Employee.find({ENAME:/^[SM]/})

24. db.Employee.find({ENAME:/il/})

25. db.Employee.find({ENAME:{$not:/a/}})

26. db.Employee.find({ENAME:/[0-9]/})

27. db.Employee.find({ENAME:/^[^aeiou]\*[aeiou][^aeiou]\*$/})(\* mean zero or more)

28. db.Employee.find({ENAME:/^[A-Z][a-z]/})

**PART-B**

1. db.Student.find({SNAME:/^K/})

2. db.Student.find({SNAME:/^[ZD]/})

3. db.Student.find({CITY:/^[A-R]/})

4. db.Student.find({SNAME:/^P.\*i$/})

5. db.Student.find({DEPARTMENT:/^C/})

6. db.Student.find({CITY:/med/},{NAME:1,SEM:1,FEES:1,DEPARTMENT:1})

7. db.Student.find({CITY:{$not:{$in:[‘Rajkot’,’Baroda’]}}},{SNAME:1,SEM:1,FEES:1,DEPARTMENT:1})

8. db.Student.find({SNAME:/^K[a-z]/})

9. db.Student.find({SNAME:/a$/})

10. db.Student.find({SNAME:/ri/})

**PART-C**

1. db.Student.find({SNAME:/^[AEIOU]/})

2. db.Student.find({$or:[{CITY:/pur$/},{CITY:/bad$/}]})

3. db.students.find({ FEES: /^1/ });

4. db.Student.find({$or:[{SNAME:/^K/},{SNAME:/^V/}]})

5. db.students.find({ CITY: { $regex: /^.{5}$/ } })

6. db.Student.find({SNAME:{$not:/e/}})

7. db.Student.find({CITY:/^Ra.\*ot$/})

8. db.Student.find({SNAME:{$regex:/^[^aeiou]\*[aeiou][^aeiou]\*$/}})

9. db.students.find({ SNAME: { $regex: /^(.).\*\1$/ } })

10. db.Student.find({ DEPARTMENT: { $regex: /^[CE]/ } })

11. db.Student.find({SNAME:/^.{5}$/})

12. db.Student.find({GENDER:'Female',CITY:/^A/})

LAB-12

PART-A

1. db.Students.aggregate([{$group:{\_id:"$DEPARTMENT"}}])

2. db.Students.aggregate([{$group:{\_id:"$CITY",count:{$sum:1}}}])

3. db.Students.aggregate([{$group:{\_id:null,totalSalary:{$sum:"$FEES"}}}])

4. db.Students.aggregate([{$group:{\_id:null,avgSalary:{$avg:"$FEES"}}}])

5. db.Students.aggregate([{$group:{\_id:null,maxFees:{$max:"$FEES"},minFees:{$min:"$FEES"}}}])

6. db.Students.aggregate([{$group:{\_id:"$CITY",totalFees:{$sum:"$FEES"}}}])

7. db.Students.aggregate([{$group:{\_id:"$GENDER",maximumFees:{$max:"$FEES"}}}])

8. db.Students.aggregate([{$group:{\_id:"$CITY",maxFees:{$max:"$FEES"},minFees:{$min:"$FEES"}}}])

9. db.Students.aggregate([{$match:{CITY:"Baroda"}},{$group:{\_id:"$CITY",count:{$sum:1}}}])

10. db.Students.aggregate([{$match:{CITY:"Rajkot"}},{$group:{\_id:"$CITY",avgSalary:{$avg:"$FEES"}}}])

11.db.Students.aggregate([{$group:{\_id:{Department:"$DEPARTMENT",Gender:"$GENDER"},count:{$sum:1}}}])

12. db.Students.aggregate([{$group:{\_id:"$DEPARTMENT",totalFees:{$sum:"$FEES"}}}])

13.db.Students.aggregate([{$group:{\_id:{Department:"$DEPARTMENT",Gender:"$GENDER"},minFees:{$min:"$FEES"}}}])

14. db.Students.aggregate([{$sort:{FEES:-1}},{$limit:5}])

15.db.Students.aggregate([{$group:{\_id:"$CITY",avgFees:{$avg:"$FEES"},count:{$sum:1}}},{$match:{count:{$gt:1}}}])

16.db.Students.aggregate([{$match:{DEPARTMENT:{$in:['Mechanical','CE']}}},{$group:{\_id:null,totalFees:{$sum:"$FEES"}}}])

17.db.Students.aggregate([{$group:{\_id:{Department:"$DEPARTMENT",Gender:"$GENDER"},count:{$sum:1}}})

18. db.Students.aggregate([

{

$match: { CITY: "Rajkot" } // Step 1: Filter students from Rajkot

},

{

$group: {

\_id: "$DEPARTMENT", // Step 2: Group by Department

averageFees: { $avg: "$FEES" } // Step 3: Calculate the average Fees

}

}

]);

19.db.Students.aggregate([{$group:{\_id:'$SEM',totalFees:{$sum:'$FEES'},averageFees:{$avg:'$FEES'}}},{$sort:{totalFees:-1}}])

20. db.students.aggregate([

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

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

{ $limit: 3 } ])

PART-B

4. db.Stock.aggregate([{$group:{\_id:'$sector',average:{$avg:'$profit'}}}])

5. db.Stock.aggregate([{$group:{\_id:'$sector',count:{$sum:1}}}])

6. db.Stock.find().sort({ pe: -1 }).limit(1);

7. db.Stock.aggregate({$match:{pe:{$gt:20}}})

8. db.Stock.aggregate([

{ $match: { sales: { $gt: 250000 } } },

{ $group: { \_id: null, totalProfit: { $sum: "$profit" } } }

]);

9. db.Stock.aggregate([{$project:{\_id:0,comapany:1,profit:1}}])

10. db.companies.aggregate([

{ $group: { \_id: null, avgEPS: { $avg: "$eps" } } },

{ $lookup: {

from: "companies",

pipeline: [

{ $match: { eps: { $gt: "$$avgEPS" } } }

],

as: "aboveAvgEPS"

} },

{ $unwind: "$aboveAvgEPS" },

{ $replaceRoot: { newRoot: "$aboveAvgEPS" } }

]);

11. db.Stock.aggregate([

{ $group: { \_id: "$sector", maxSales: { $max: "$sales" } } }

])

12. db.Stock.aggregate([

{ $group: {

\_id: "$sector",

totalSales: { $sum: "$sales" },

totalProfit: { $sum: "$profit" }

} }

]);

13. db.Stock.aggregate([

{ $sort: { profit: -1 } }

]);

14. db. Stock.aggregate([

{ $group: { \_id: null, avgROE: { $avg: "$roe" } } }

]);

15. db.Stock.aggregate([

{ $group: {

\_id: "$sector",

minEPS: { $min: "$eps" },

maxEPS: { $max: "$eps" }

} }

]);

PART-C

1. db.Stock.aggregate([

{ $match: { profit: { $gt: 30000 } } },

{ $count: "count" }]);

2. db. Stock.aggregate([

{ $group: { \_id: "$sector", totalProfit: { $sum: "$profit" } } },

{ $sort: { totalProfit: -1 } }

]);

3. db. Stock.aggregate([

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

{ $limit: 3 }

]);

4. db. Stock.aggregate([

{ $group: { \_id: "$sector", avgPE: { $avg: "$pe" } } }

]);

5. db. Stock.aggregate([

{ $project: { \_id: 0, company: 1, totalSales: "$sales", totalProfit: "$profit" } }

]);

6. db. Stock.aggregate([

{ $match: { sales: { $lt: 400000 } } },

{ $sort: { sales: 1 } }

]);

7. db. Stock.aggregate([

{ $group: { \_id: "$sector", companyCount: { $sum: 1 } } }

]);

8. db. Stock.aggregate([

{ $match: { sales: { $gt: 200000 } } },

{ $group: { \_id: null, avgROE: { $avg: "$roe" } } }

]);

9. db. Stock.aggregate([

{ $group: { \_id: "$sector", maxProfit: { $max: "$profit" } } }

]);

10. db. Stock.aggregate([

{ $group: { \_id: "$sector", totalSales: { $sum: "$sales" }, companyCount: { $sum: 1 } } }

]);

11. db. Stock.aggregate([

{ $match: { profit: { $gt: 20000 } } },

{ $project: { \_id: 0, company: 1, profit: 1 } }

]);

12. db. Stock.aggregate([

{ $sort: { roe: 1 } }

]);

LAB-13

PART-A

1. db.Stock.createIndex({company:1})

2. db.Stock.createIndex({sector:1,sales:1})

3. db.Stock.getIndexes()

4. db.Stock.dropIndex('Company\_1')

5. const cursor=db.Stock.find();

cursor.forEach(doc=>printjson(doc));

6. const cursor=db.Stock.find().limit(3);

cursor.forEach(doc=>printjson(doc));

7. const cursor\_sorted\_onsales=db.Stock.find().sort({sales:-1})

const cursor=db.Stock.find()

cursor.forEach(cursor\_sorted\_onsales=>printjson(cursor\_sorted\_onsales));

8. const cursor=db.Stock.find().skip(2);

cursor.forEach(doc=>printjson(doc));

9. const allDocsArray=db.Stock.find().toArray()

printjson(allDocsArray)

10. db.createCollection("Companies",{validator:{

$jsonSchema:{bsonType:"object",

required:["company","sector"],

properties:{

comapny:{bsonType:'string'},

sector:{bsonType:'string'}

}

}

}})

PART-B

1. db.createCollection("scripts",{validator:{

$jsonSchema:{bsonType:"object",

required:['eps','pe','roe'],

properties:{

eps:{bsonType:'number'},pe:{bsonType:'number'},roe:{bsonType:'number’}

}

}

}})

2. db.Products.insertMany([

{

name: "Iphone 16 Pro Max",

price: 100000,

manufacturer: {

name: "Apple",

country: "USA",

establishedYear: 1998

},

categories: ["Electronics", "Mobile", "Smart Devices"]

},

{

name: "Laptop Pro",

price: 129900,

manufacturer: {

name: "Ishita",

country: "Germany",

establishedYear: 2005

},

categories: ["Electronics", "Computers"]

}

]);

PART-C

1. db.createCollection("financial\_Reports", {

validator: {

$jsonSchema: {

bsonType: "object",

required: ["revenue"],

properties: {

revenue: {

bsonType: "double",

minimum: 0.01,

description: "Required field: Revenue must be a positive number."

},

expenses: {

bsonType: ["double", "null"],

description: "Optional field: Expenses must be a number."

},

netIncome: {

bsonType: ["double", "null"],

description: "Optional field: Net Income must be a number."

}}}}

});

2. db.Student.insertMany([

{

"name": "John Doe",

"address": {

"street": "123 Main St",

"city": "New York",

"state": "NY",

"zip": "10001"

},

"mobilenumber": ["1234567890", "9876543210"],

"emailaddress": ["john.doe@example.com", "johndoe123@gmail.com"]

},

{

"name": "Alice Smith",

"address": {

"street": "456 Elm St",

"city": "Los Angeles",

"state": "CA",

"zip": "90001"

},

"mobilenumber": ["1112223333", "4445556666"],

"emailaddress": ["alice.smith@example.com", "alice\_smith@yahoo.com"]

}

])