**PROGRAM -1**

**AIM:** To build a sample collections/documents to perform query operations.

Create database college and collection students & insert student details into it.

**CODE**:

>use college

>db.createCollection(“students”)

>db.students.insert({SRN:'101',SName:'Stefi',Degree:'MCA',Sem:2,CGPA:8.9})

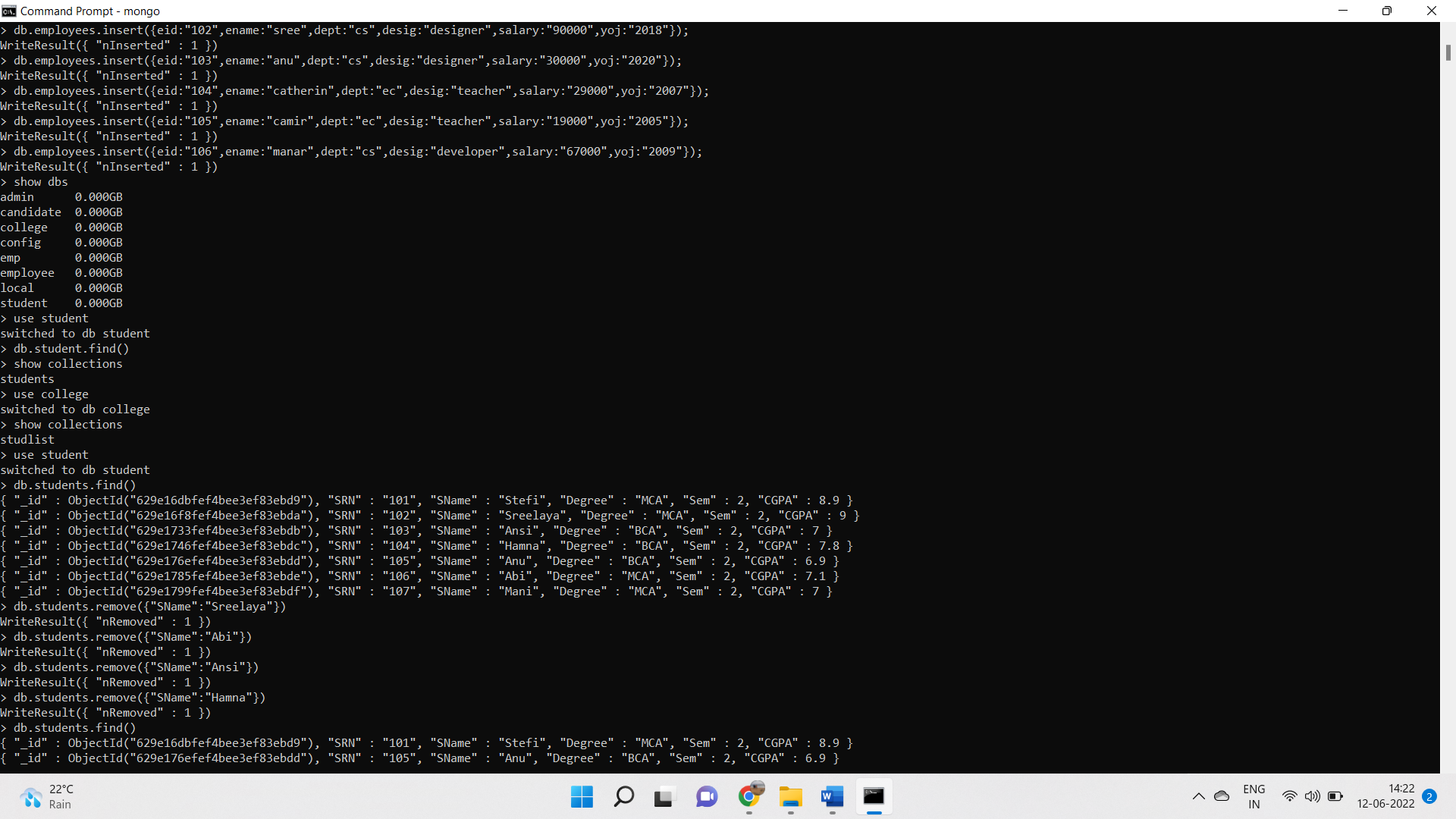
>db.students.insert({SRN:'102',SName:'Sreelaya',Degree:'MCA',Sem:2,CGPA:9})

> db.students.insert({SRN:'103',SName:'Ansi',Degree:'BCA',Sem:2,CGPA:7})

>db.students.insert({SRN:'104',SName:'Hamna',Degree:'BCA',Sem:2,CGPA:7.8})

> db.students.insert({SRN:'105',SName:'Anu',Degree:'BCA',Sem:2,CGPA:6.9})

>db.students.find()



**PROGRAM -2**

**AIM:** Create a Database ‘Student’ with the fields SRN, SName, degree, semester, CGPA and create a collection ‘Students’.

1. Display all the documents.
2. Display all the students in BCA.
3. Display all the students in ascending order
4. Display all the first five students.
5. Display students 5,6,7
6. Display the degree of student ‘Anu’.
7. Display student details of 5,6,7 in descending order of percentage.
8. Display the number of students in BCA
9. Display all the degrees without "\_id"
10. Display the distinct degrees.
11. Display all the BCA students with CGPA>6 but less than 7.1
12. Display all the BCA students and in 2nd sem.

**CODE**:

> use student;

> db.createCollection("students")

>db.students.insert({SRN:'101',SName:'Stefi',Degree:'MCA',Sem:2,CGPA:8.9})

>db.students.insert({SRN:'102',SName:'Sreelaya',Degree:'MCA',Sem:2,CGPA:9})

> db.students.insert({SRN:'103',SName:'Ansi',Degree:'BCA',Sem:2,CGPA:7})

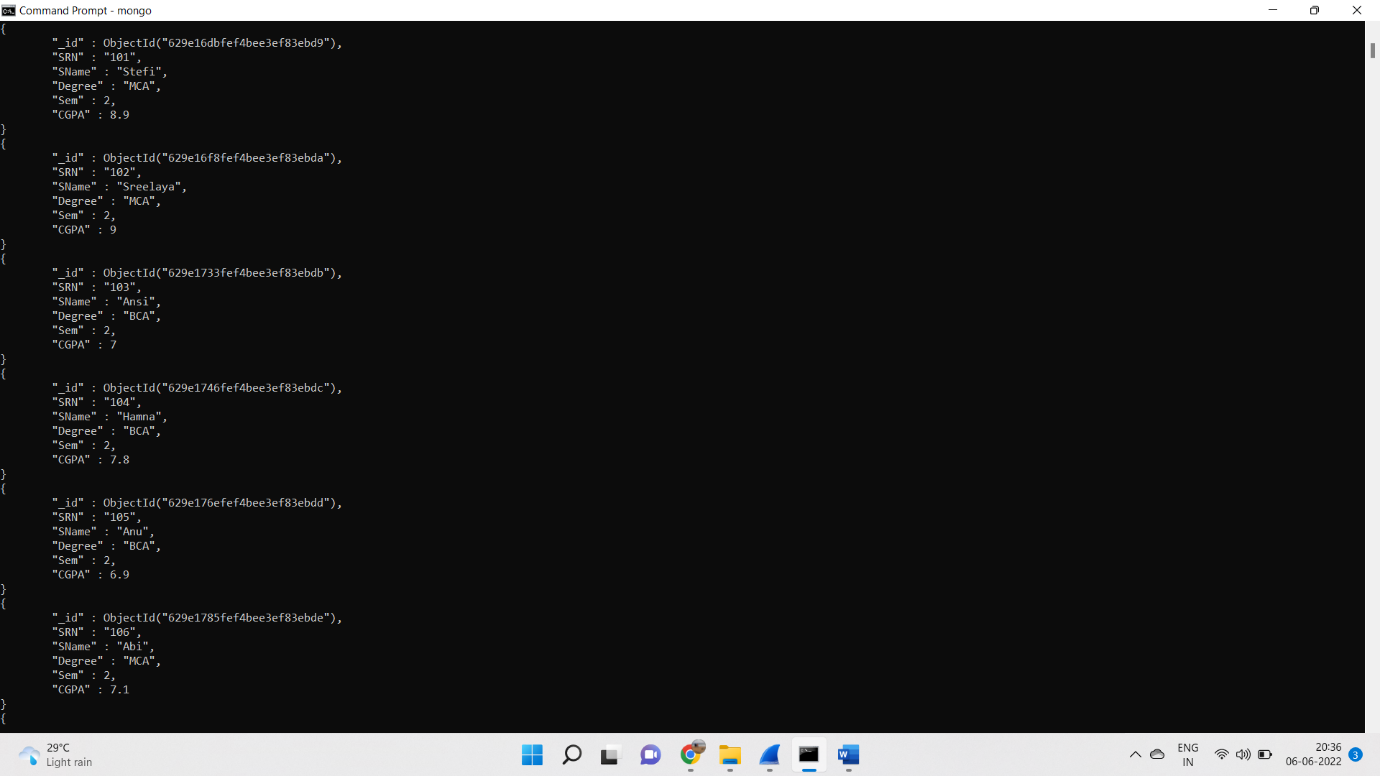
>db.students.insert({SRN:'104',SName:'Hamna',Degree:'BCA',Sem:2,CGPA:7.8})

> db.students.insert({SRN:'105',SName:'Anu',Degree:'BCA',Sem:2,CGPA:6.9})

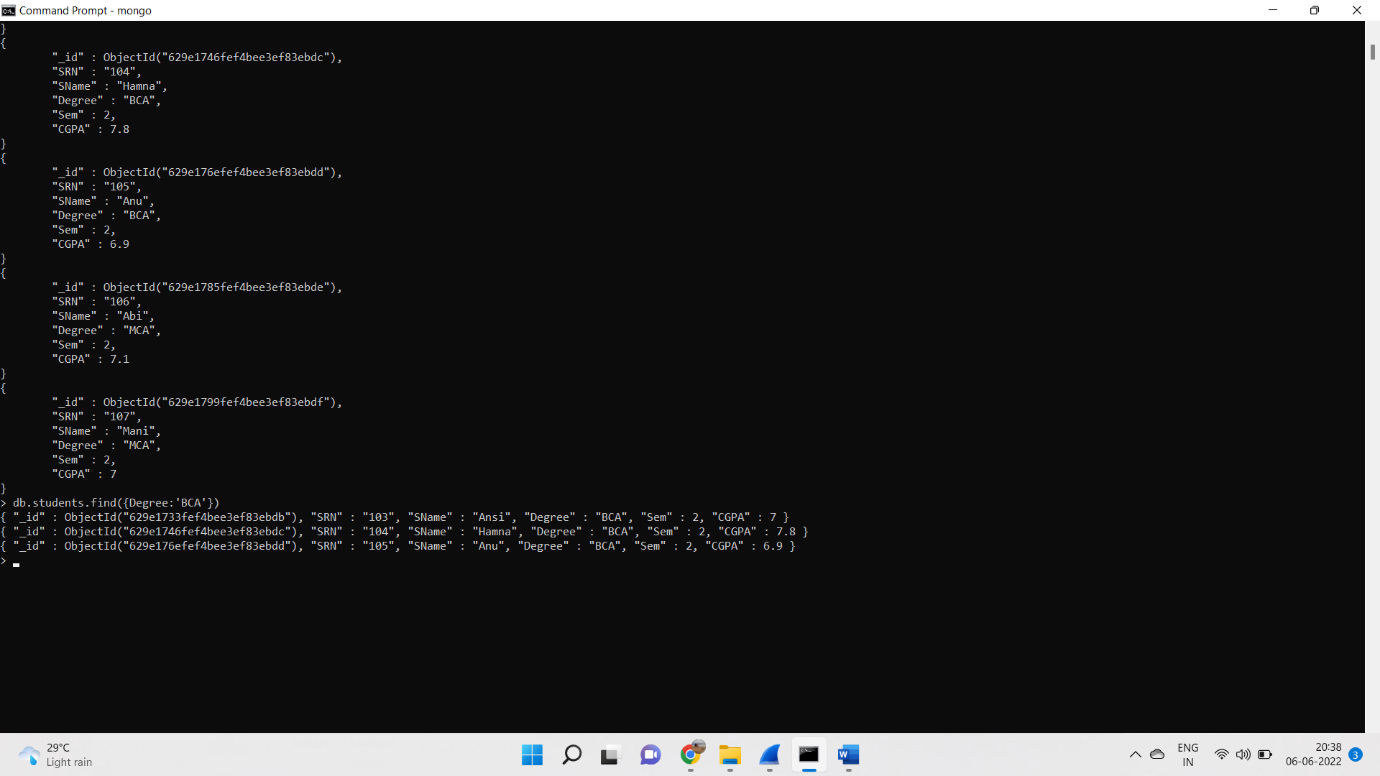
> db.students.insert({SRN:'106',SName:'Abi',Degree:'MCA',Sem:2,CGPA:7.1})

> db.students.insert({SRN:'107',SName:'Mani',Degree:'MCA',Sem:2,CGPA:7})

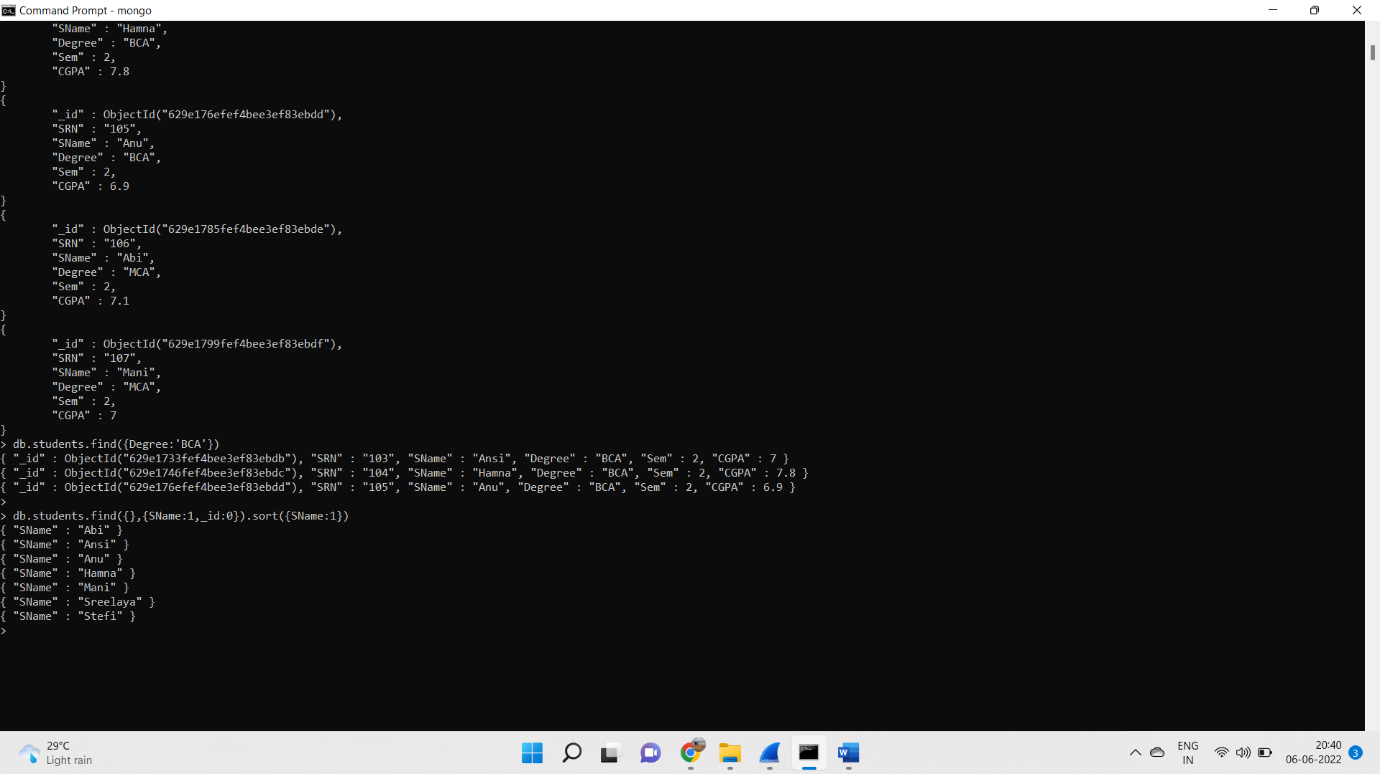
1)db.students.find().pretty()



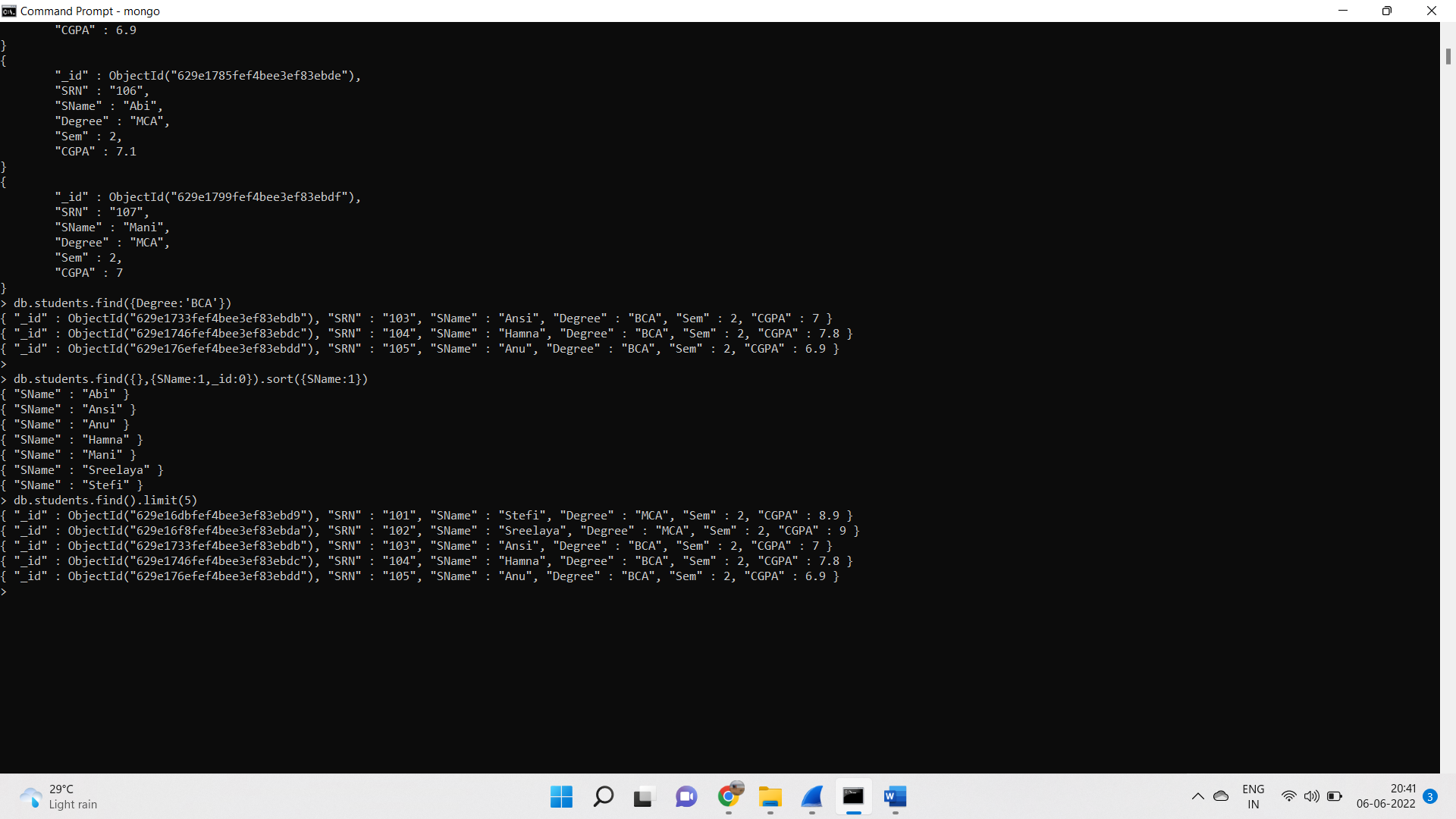
2) db.students.find({Degree:'BCA'})



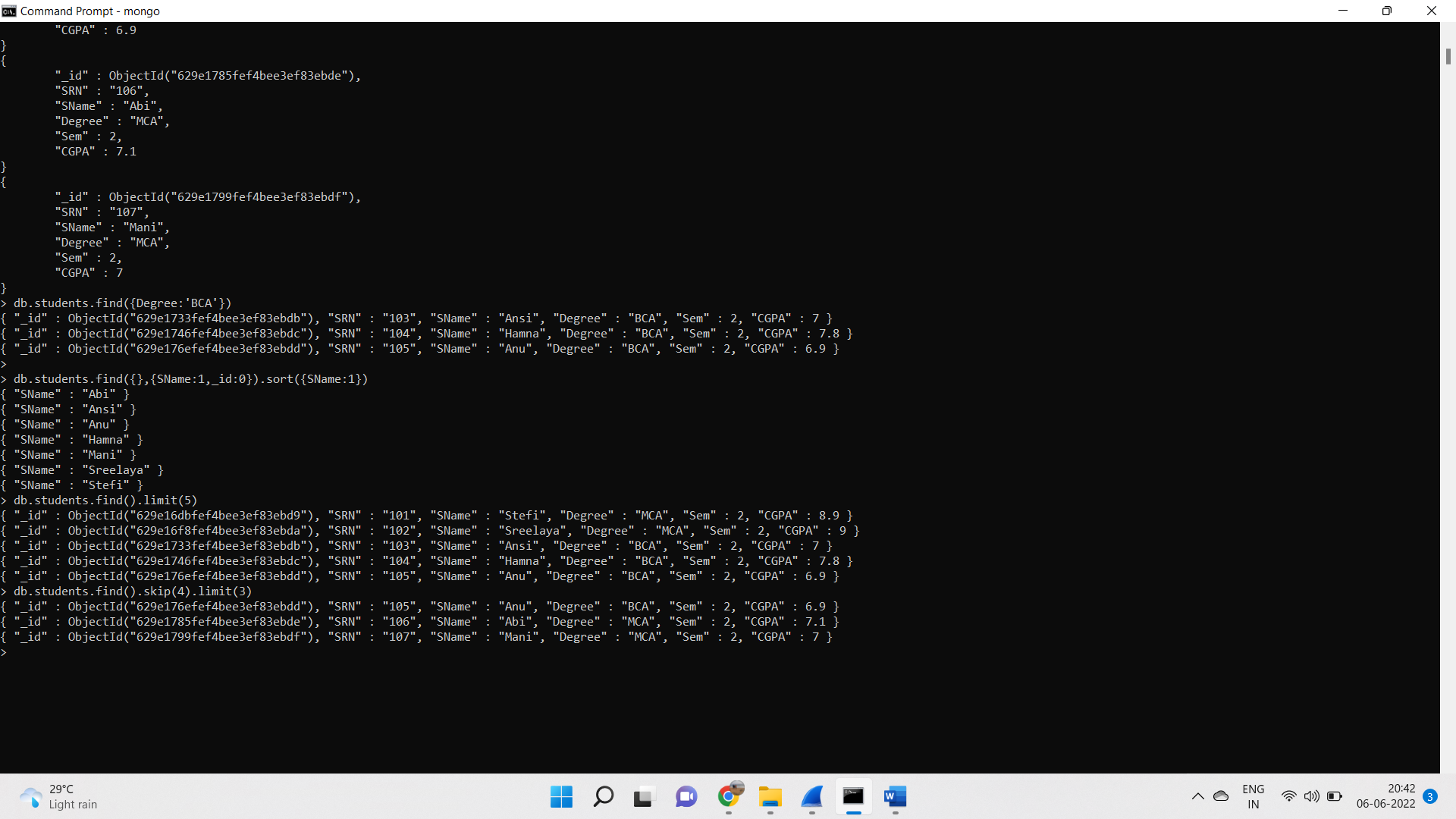
3) db.students.find({},{SName:1,\_id:0}).sort({SName:1})



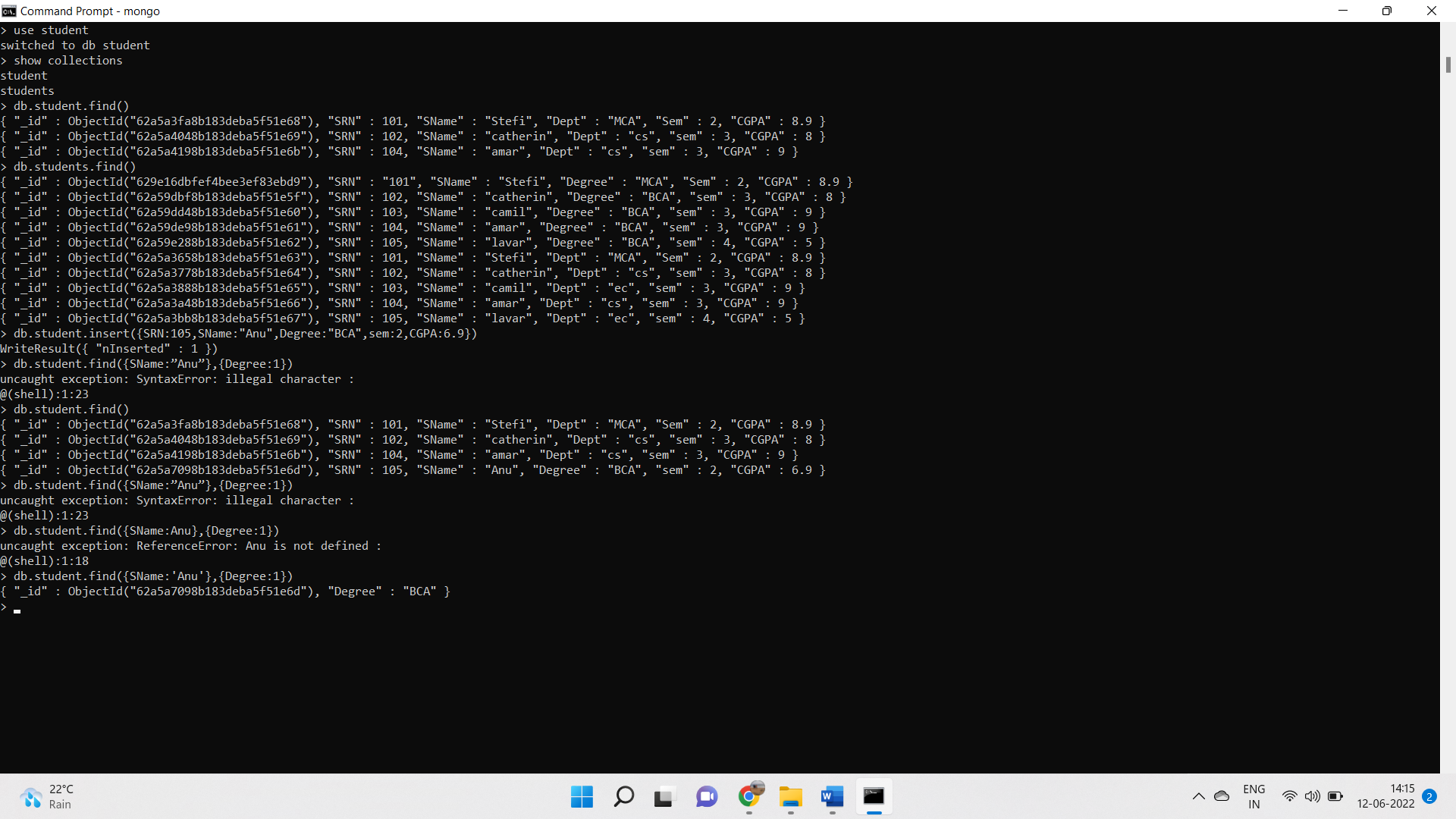
4) db.students.find().limit(5)



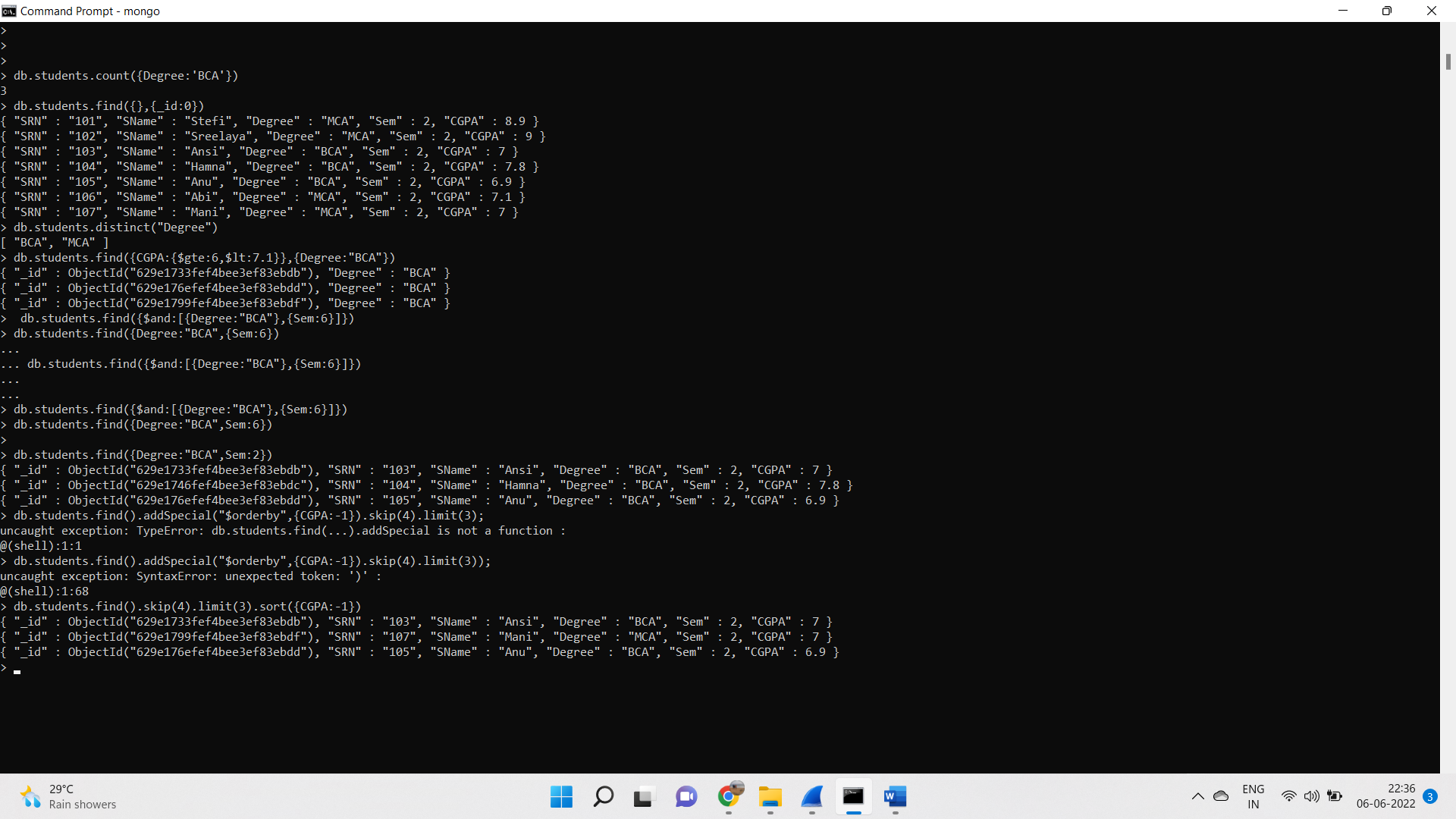
5) db.students.find().skip(4).limit(3)



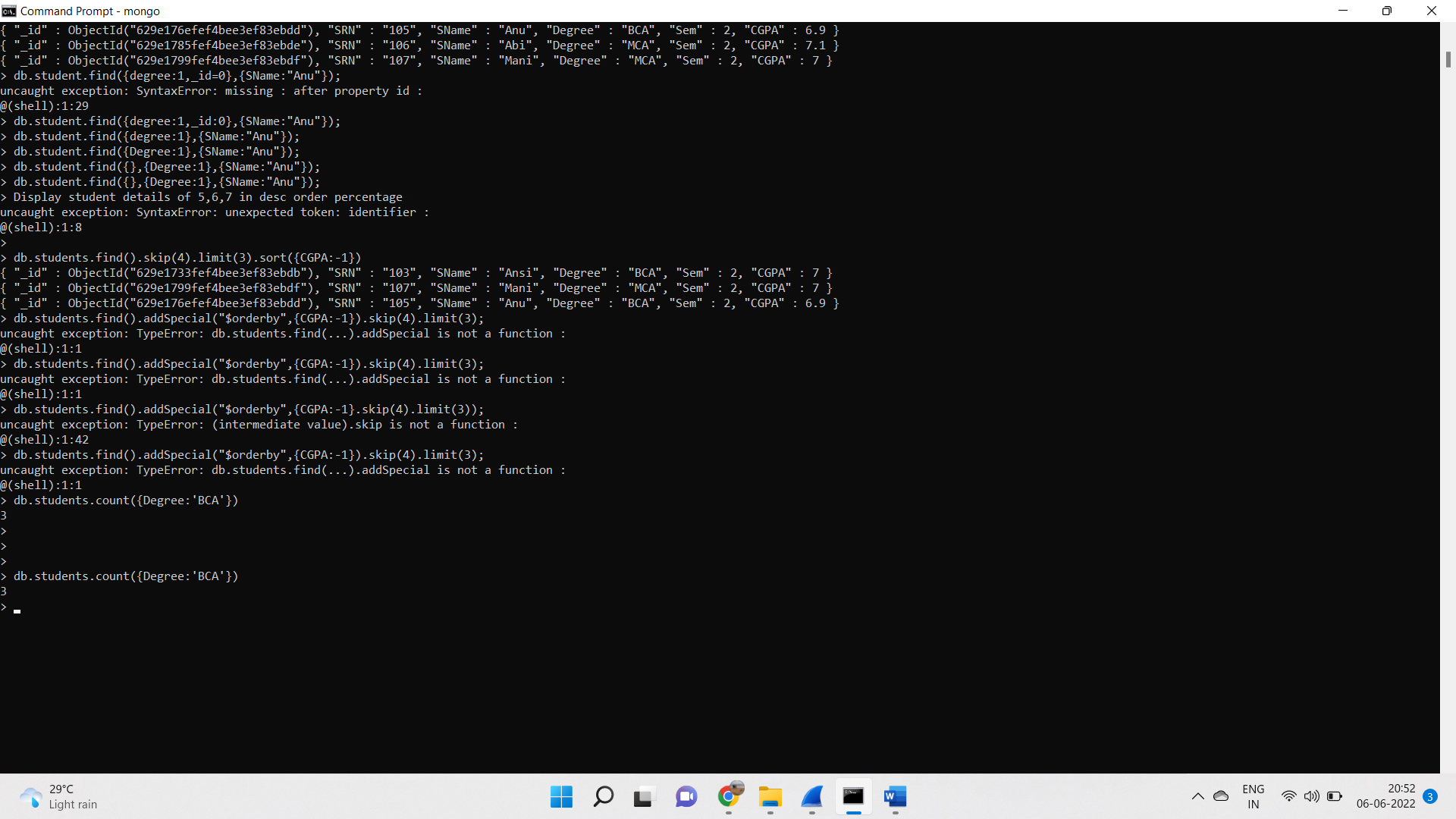
6)db.student.find({SName:”Anu”},{Degree:1})



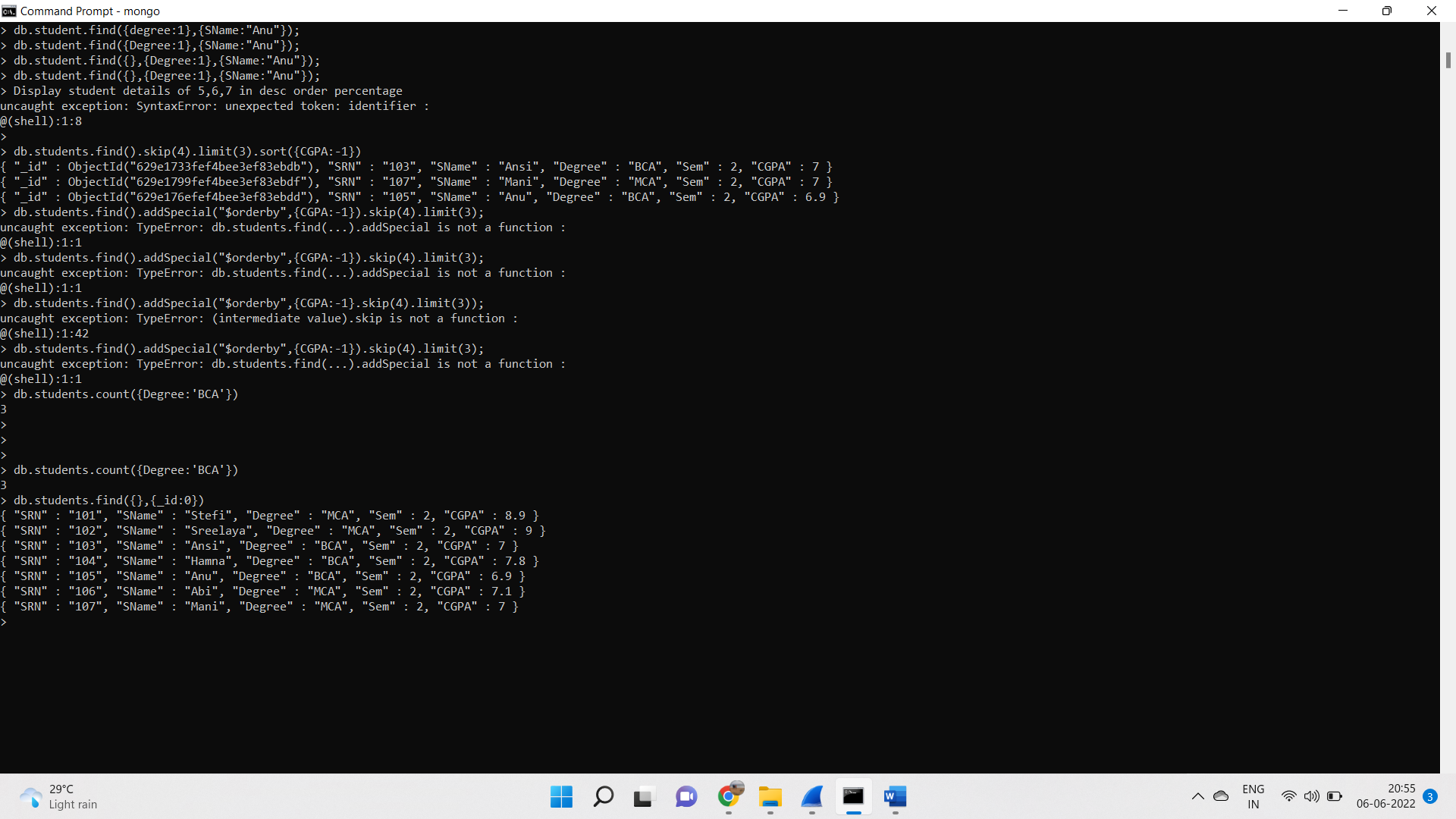
7) db.students.find().skip(4).limit(3).sort({CGPA:-1})



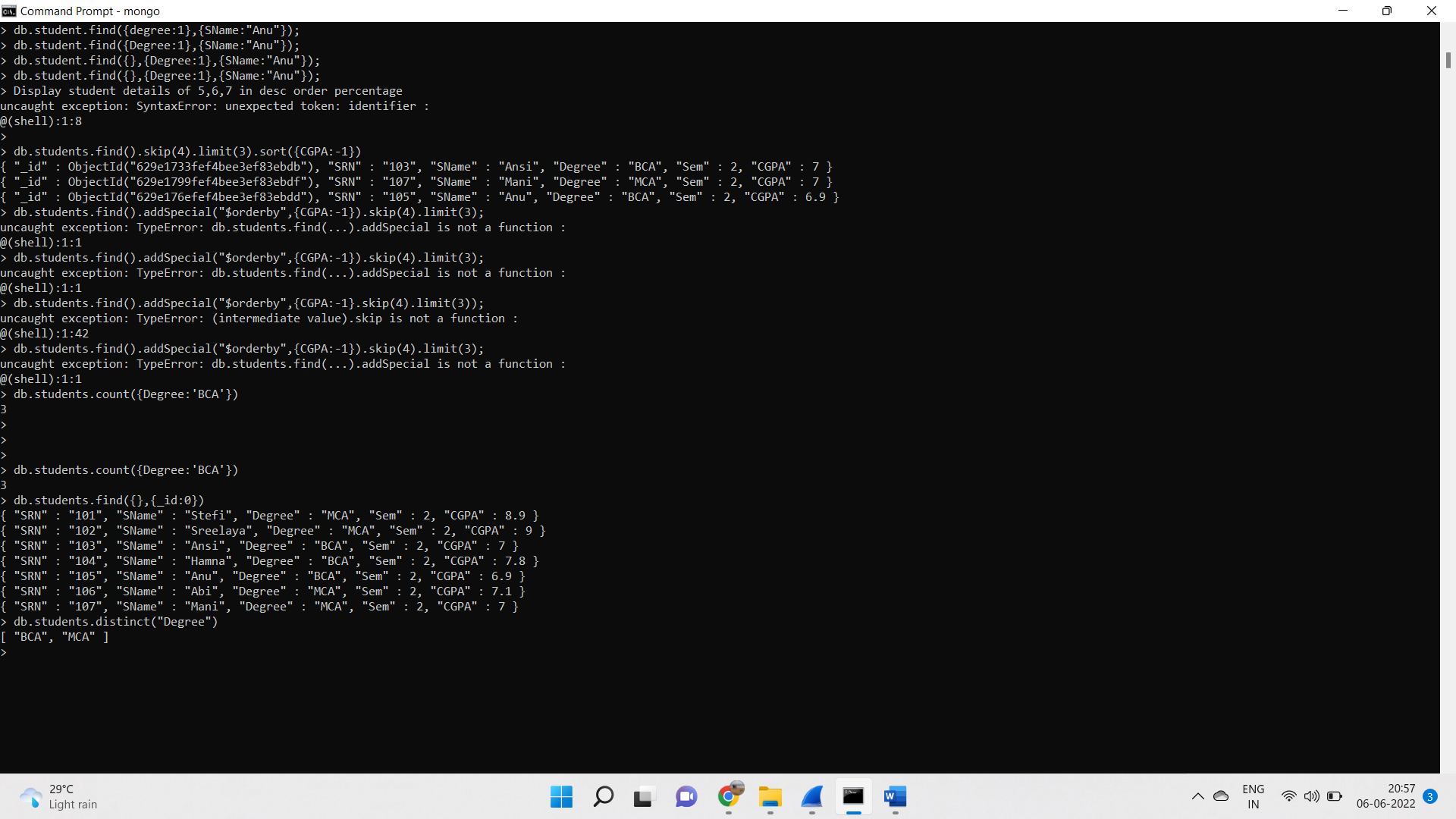
|  |
| --- |
|  |
| 8)db.students.count({Degree:'BCA'}) |
|  |



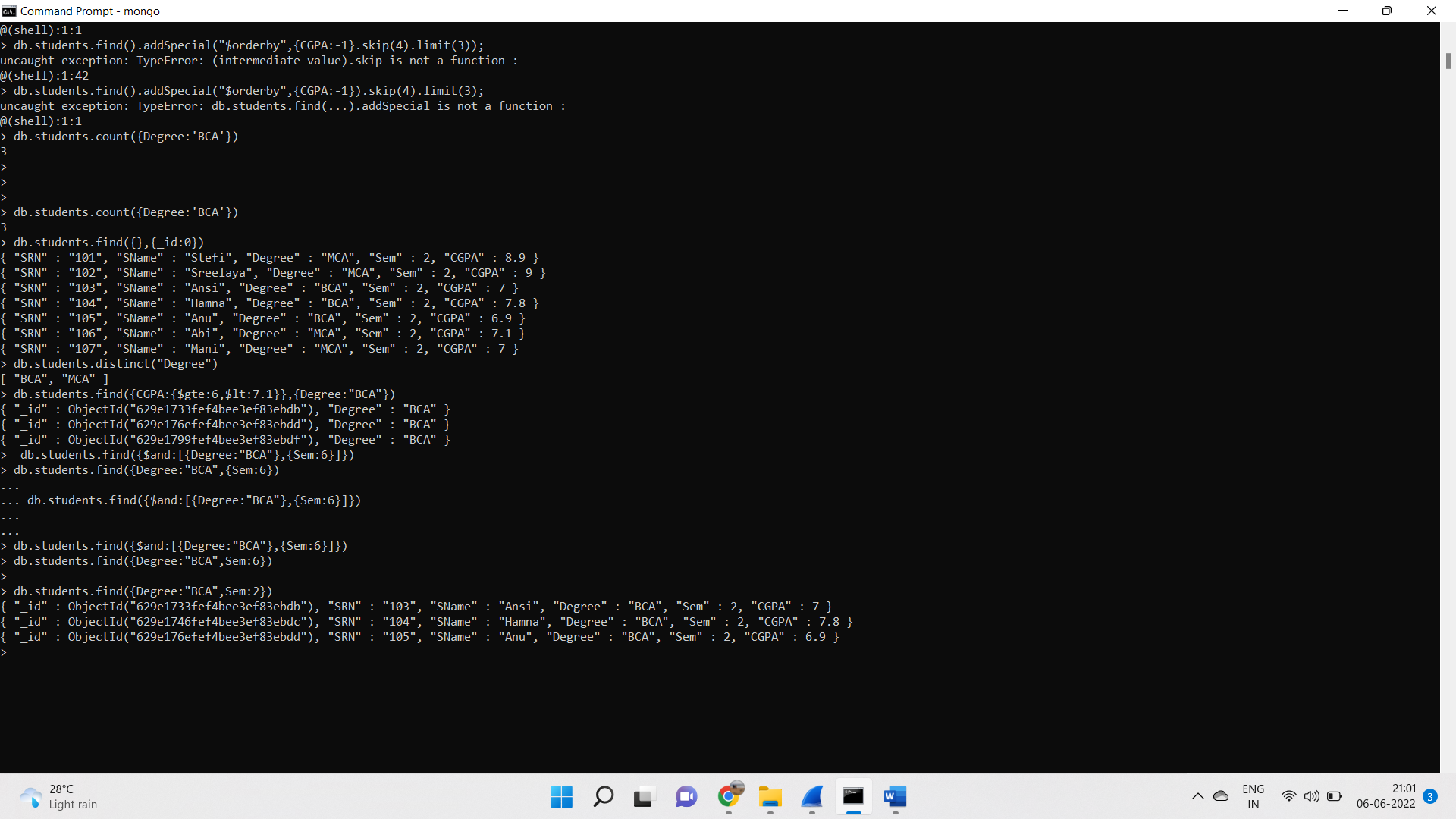
9) db.students.find({},{\_id:0})



10) db.students.distinct("Degree")



12)db.students.find({Degree:"BCA",Sem:2})



**PROGRAM-3**

**AIM**:Create an employee database with the fields: {eid, ename, dept, desig, salary, yoj, address {dno,street,locality,city}}

1) Display all the employees with salary in the range(50000,75000).

2) Display all the employees with design developer

3) Display the salary of Rahul

4) Display the city of employee.

5) Update the salary of developers by 5000.

6) Add field age to employee.

7) Remove yoj from Rahul

8) Add an array field project to Rahul..

9) Add p2 and p3 project to Rahul

10) Remove p3 from Rahul.

11) Add a new embedded object “contacts” with “email” and “phone” as array objects to Rahul.

**CODE:**

> use emp

> db.emp.createCollection("employee")

>db.emp.insert({eid:"101",ename:"manoj",dept:"hr",desig:"developer",salary:"60000",yoj:"2020",address:[{"street":"karikode","locality":"karikode","city":"kollam"}]});

>db.emp.insert({eid:"102",ename:"stefi",dept:"hr",desig:"designer",salary:"65000",yoj:"2012",address:[{"street":"tvm","locality":"pattam","city":"tvm"}]});

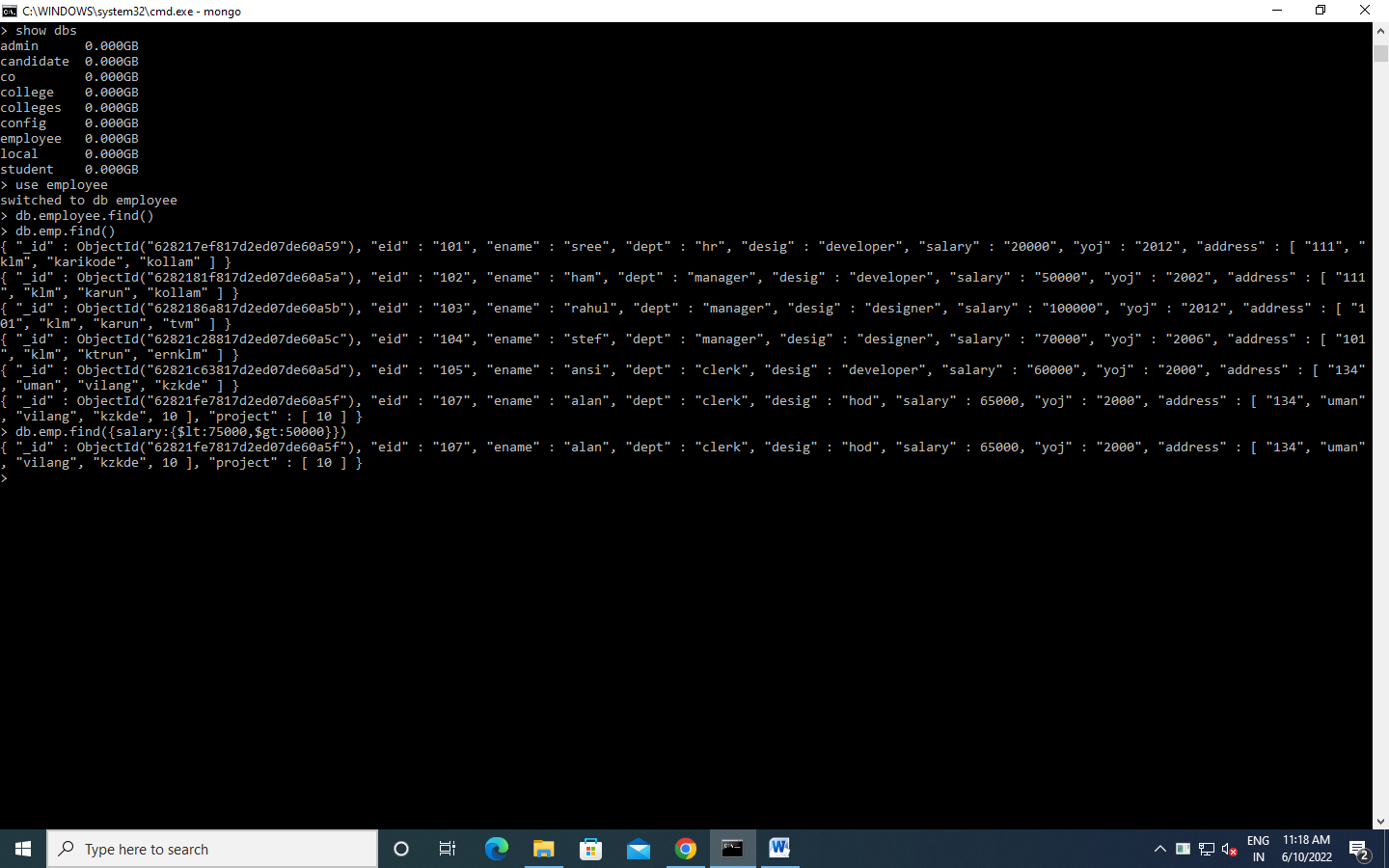
>db.emp.insert({eid:"103",ename:"sree",dept:"manager",desig:"developer",salary:"15000",yoj:"2021",address:[{"street":"kochi","locality":"kochi","city":"ernakulam"}]});

>db.emp.insert({eid:"104",ename:"rahul",dept:"manager",desig:"designer",salary:"9000",yoj:"2012",address:[{"street":"karikode","locality":"karikode","city":"kollam"}]});

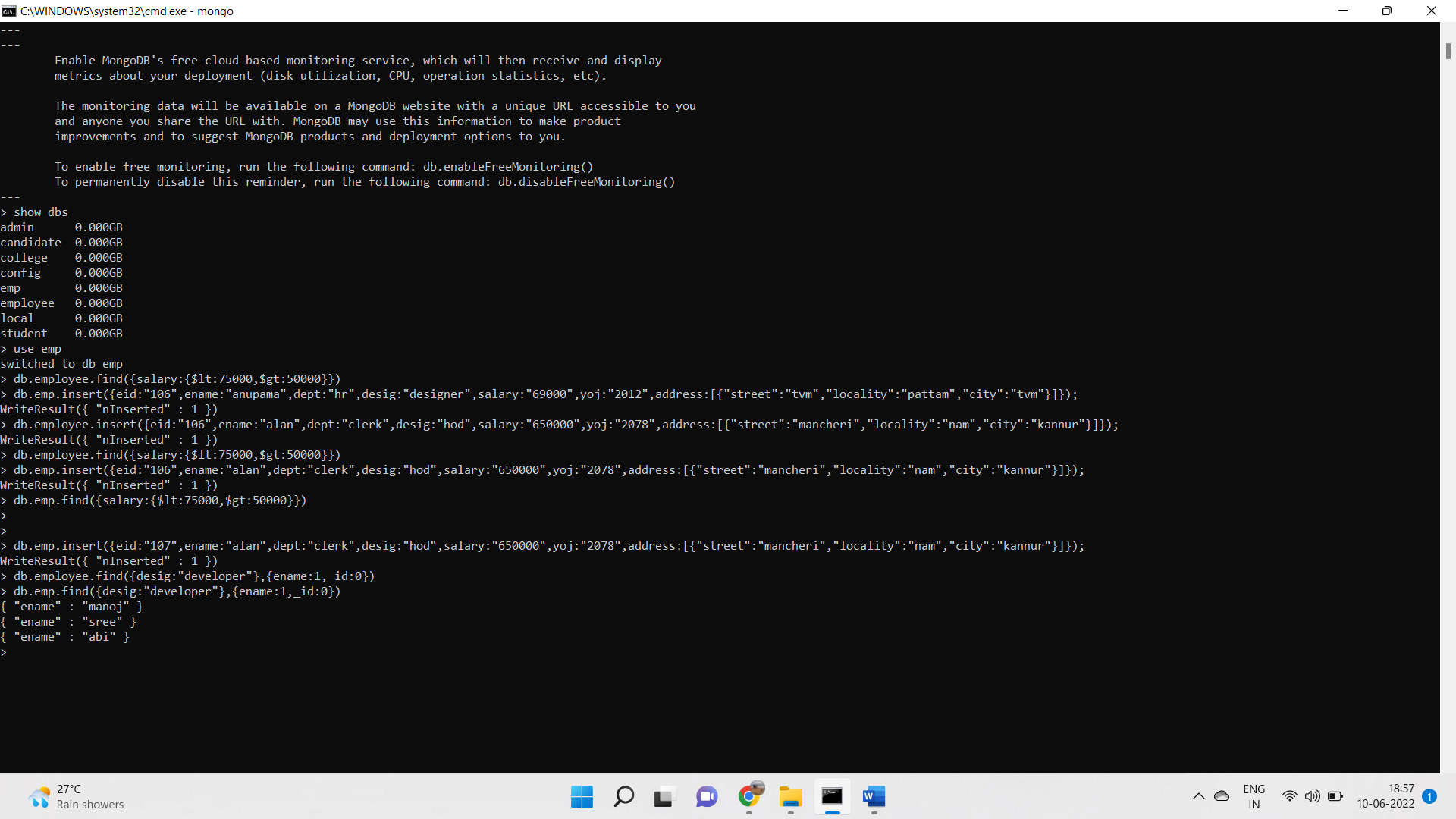
>db.emp.insert({eid:"105",ename:"abi",dept:"manager",desig:"developer",salary:"95000",yoj:"2000",address:[{"street":"tvm","locality":"thambanoor","city":"tvm"}]});

>db.emp.insert({eid:"107",ename:"alan",dept:"clerk",desig:"hod",salary:"65000",yoj:"2000",address:[{"street":"uman","locality":”vilang","city":"kzkd"}]});

1)db.employee.find({salary:{$lt:75000,$gt:50000}})



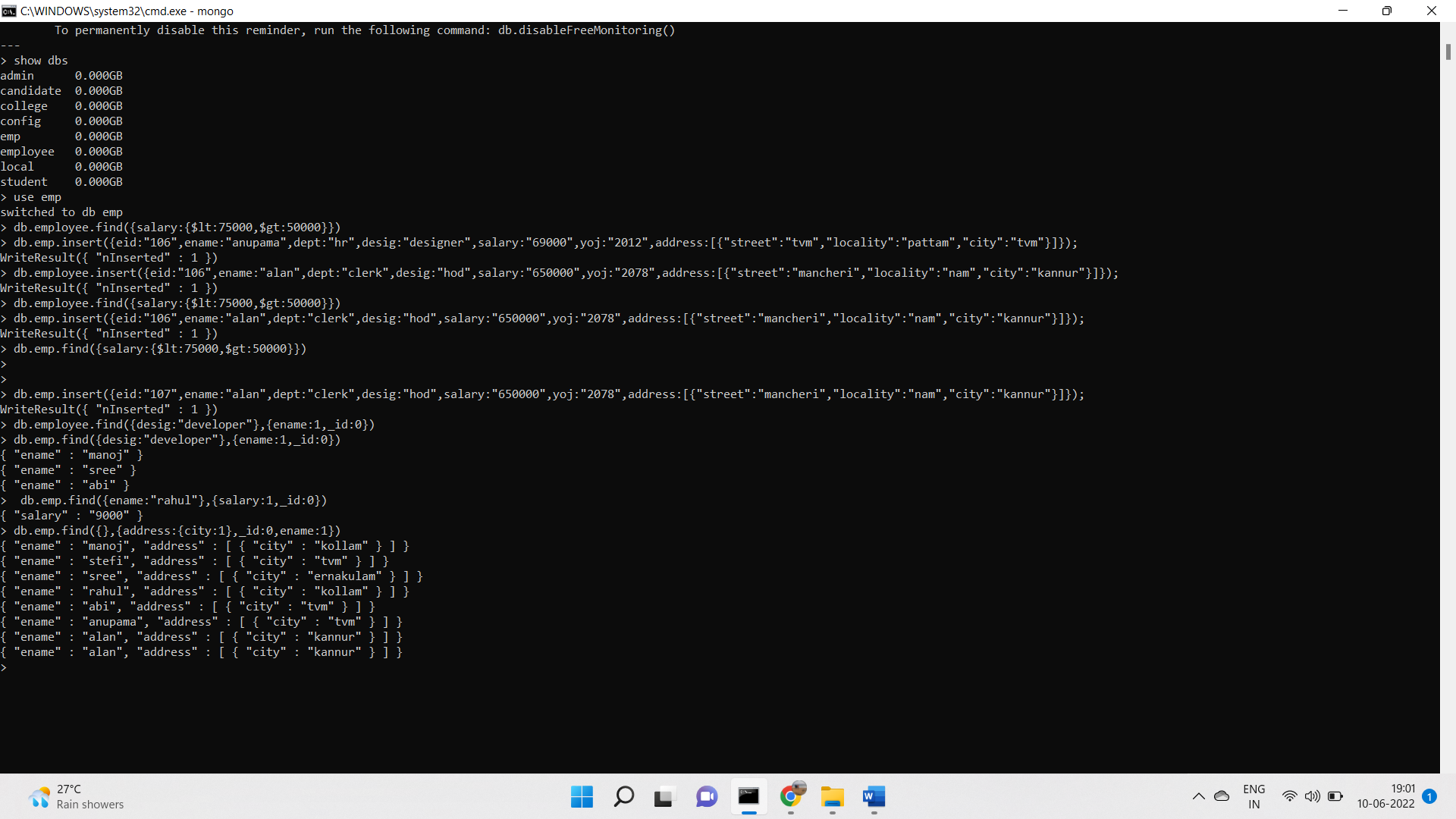
2) db.employee.find({desig:"developer"},{ename:1,\_id:0})



3) db.employee.find({ename:"rahul"},{salary:1,\_id:0})

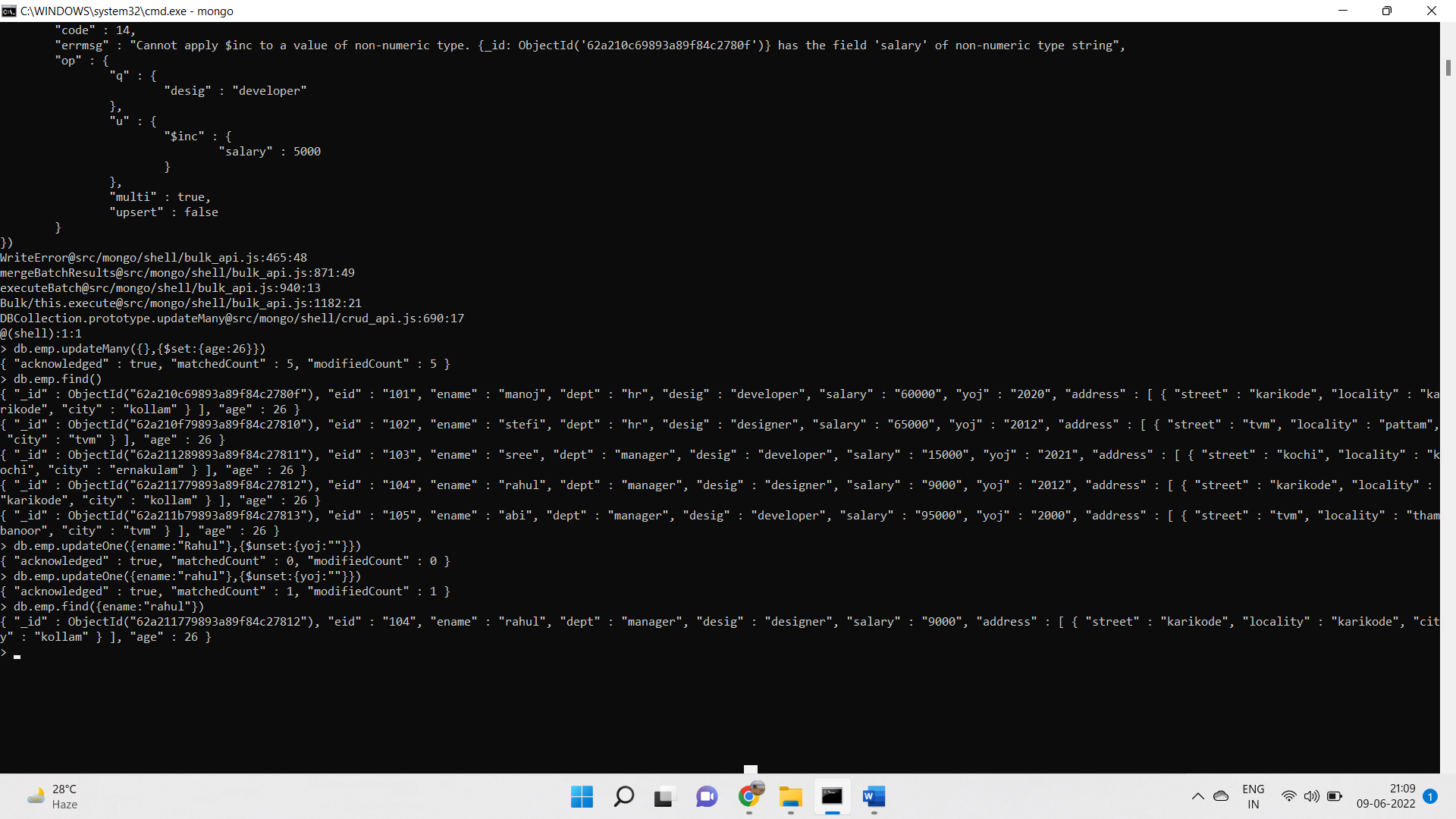


4) db.employee.find({},{address:{city:1},\_id:0,ename:1})

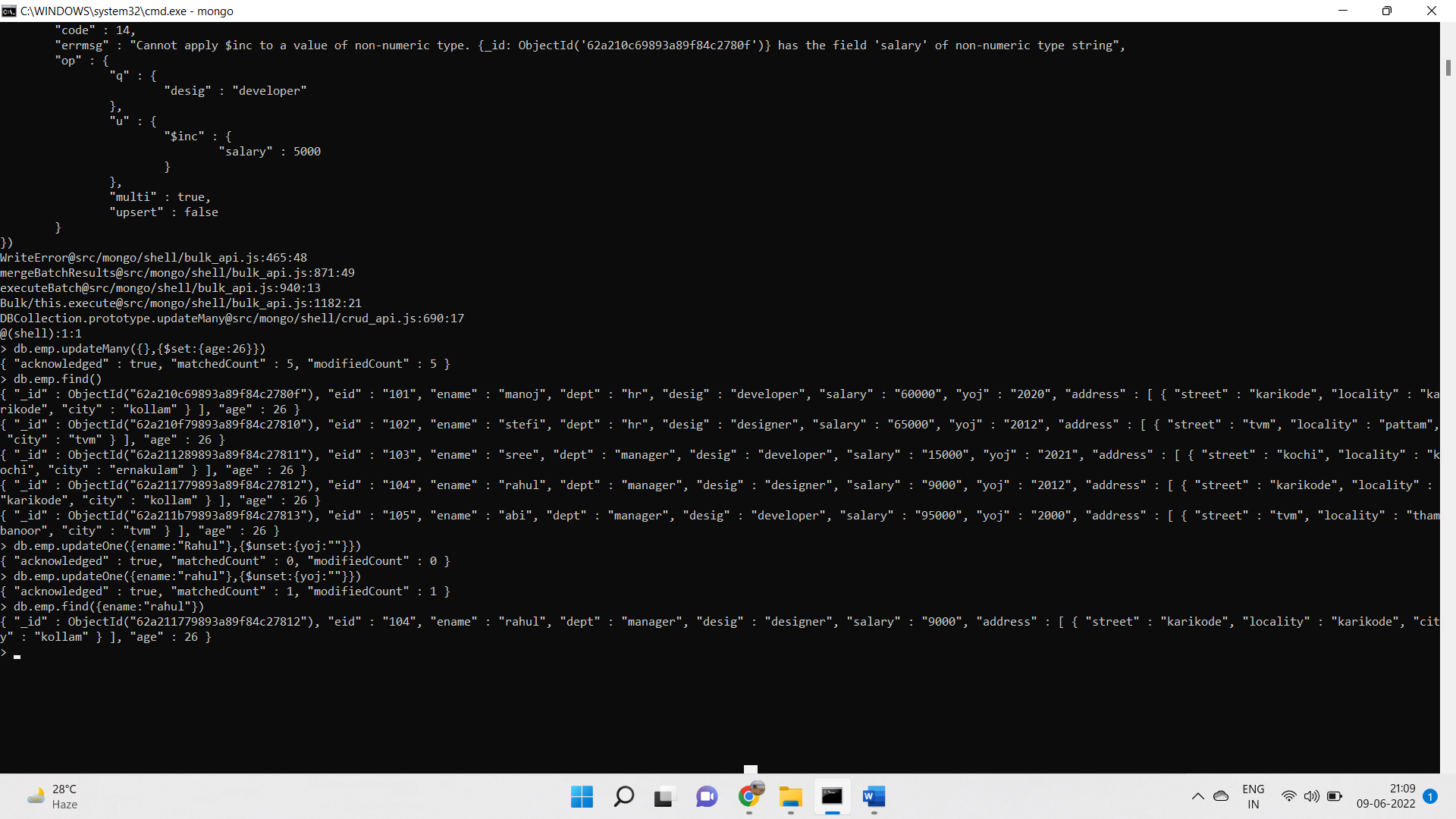


5) db.employee.updateMany({desig:"developer"},{$inc:{salary:5000}})

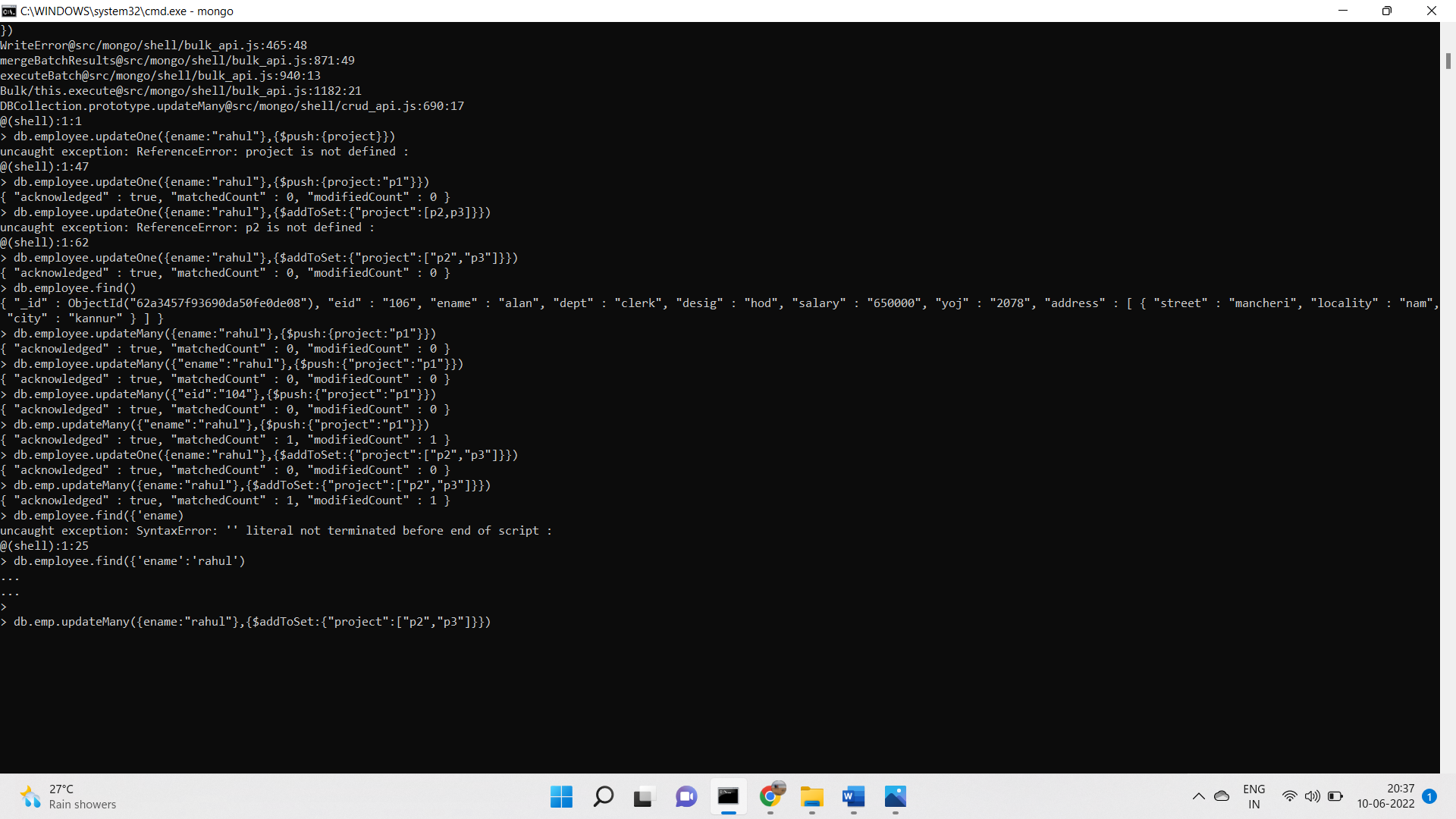
6) db.employee.updateMany({},{$set:{age:26}})

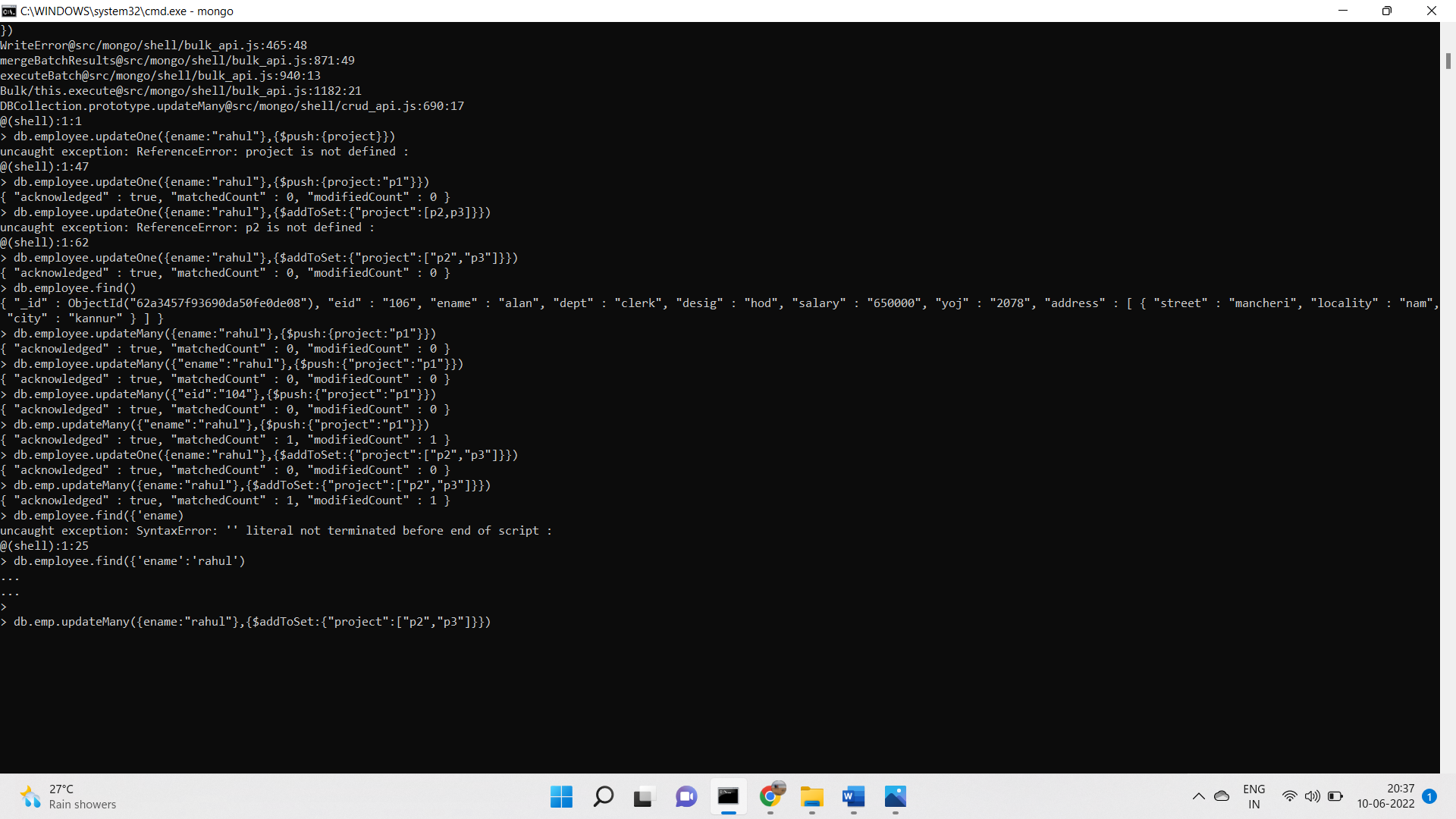


7) db.employee.updateOne({ename:"rahul"},{$unset:{yoj:""}})

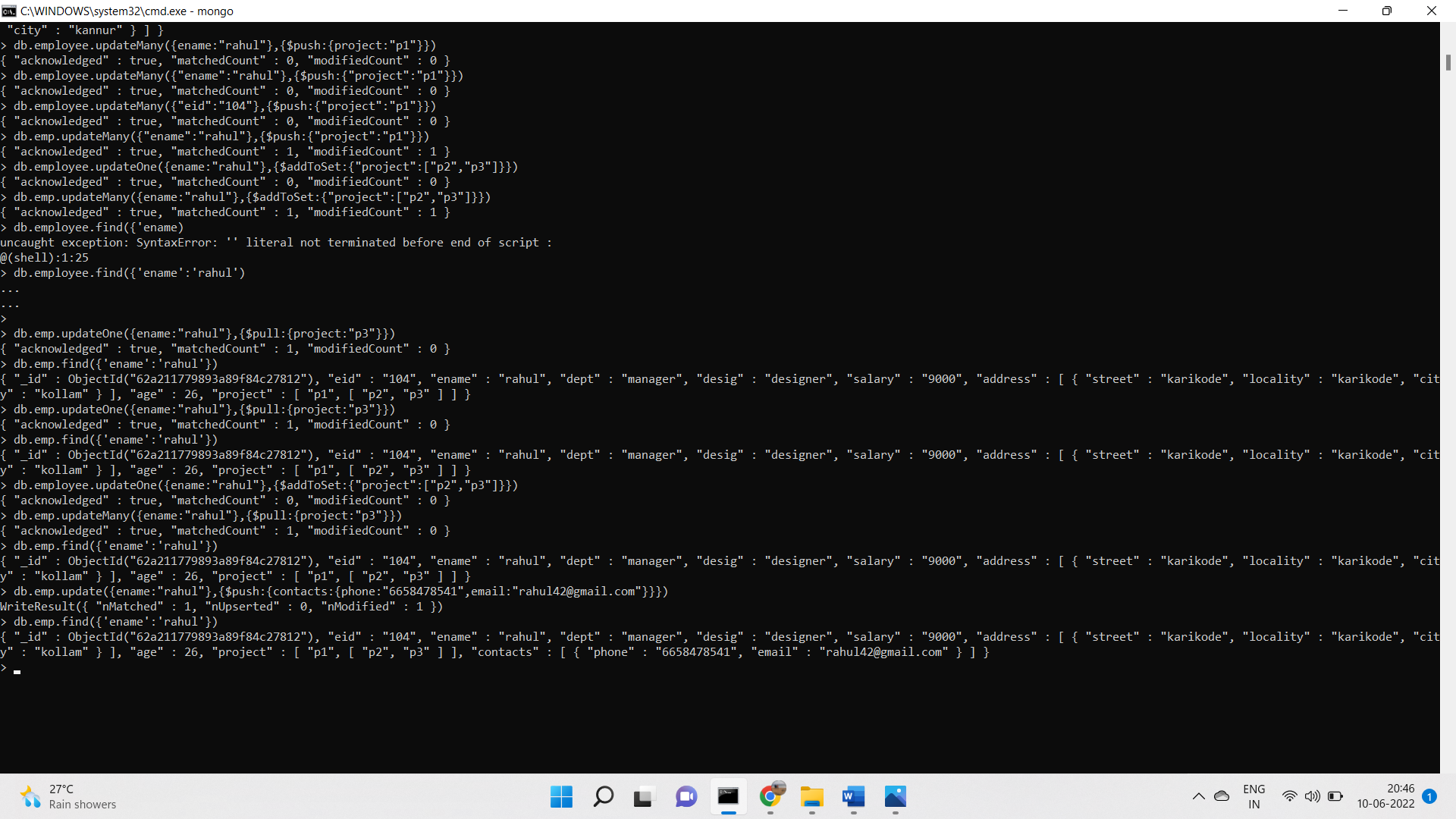


8) db.employee.updateMany({"ename":"rahul"},{$push:{"project":"p1"}})



9)db.employee.updateMany({ename:"rahul"},{$addToSet:{"project":["p2","p3"]}})

10)db.employee.updateOne({ename:"rahul"},{$pull:{project:"p3"}})

11)db.employee.update({ename:"Rahul"},{$push:{contacts:{phone:"3214120311",email:"rahulkrishna342@gmail.com"}}})

**PROGRAM -4**

**AIM:** Create a database named college and then create a collection named students. Insert some values into it. Write a MongoDB Query to:

1. Display details of students who have their name starting with the letter ‘C’ using $regex operator
2. Display details of students who have their name ending with the letter ‘r’ using $regex operator
3. Display details of students who are having ‘CS’ as their department using $regex operator
4. Remove details of student who are having ‘EC’ as their department

**CODE:**

>use college

>db.createCollection(“students”)

>db.student.insert({SRN:101, SName : Stefi, Dept : MCA, Sem: 2, CGPA: 8.9})

>db.student.insert({SRN:102,SName:"catherin",Dept:"cs",sem:3,CGPA:8})

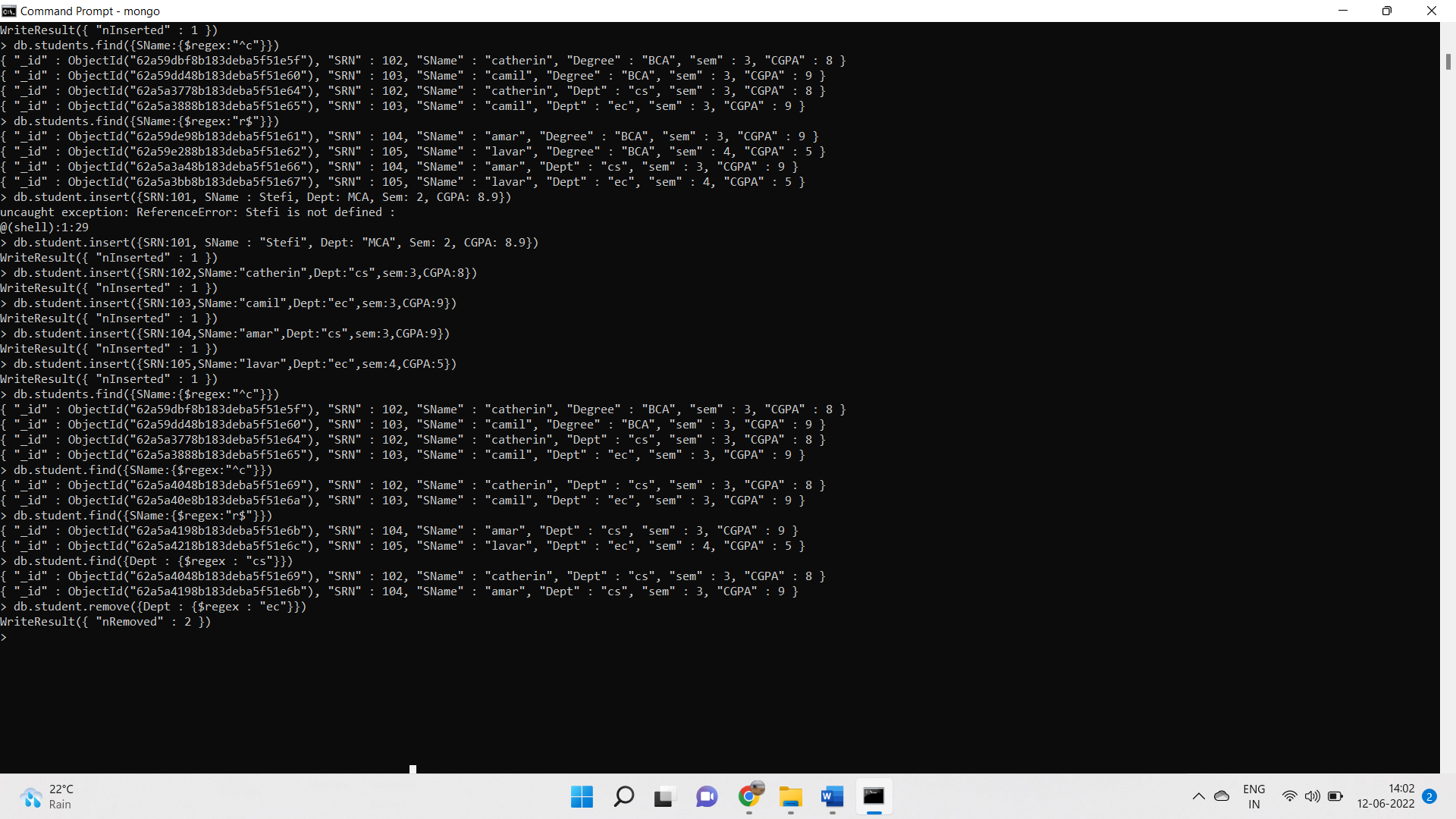
>db.student.insert({SRN:103,SName:"camil",Dept:"ec",sem:3,CGPA:9})

> db.student.insert({SRN:104,SName:"amar",Dept:"cs",sem:3,CGPA:9})

> db.student.insert({SRN:105,SName:"lavar",Dept:"ec",sem:4,CGPA:5})

1. db.student.find({SName:{$regex:"^c"}})
2. db.student.find({SName:{$regex:"r$"}})
3. db.student.find({Dept : {$regex : "cs"}})
4. db.student.remove({Dept : {$regex : "ec"}})

**OUTPUT:**



**PROGRAM -5**

**AIM:** Create database ‘candidate’ and collection ‘details’.

1)Query customer who are either male or younger than 25?

2)Calculate total purchase amount for males and females using aggregate method

3)Select customers who are older than 25 and calculate the average purchase amount for males and females

4)sort the data based on average amount.

**CODE:**

> use candidate

> db.createCollection("details")

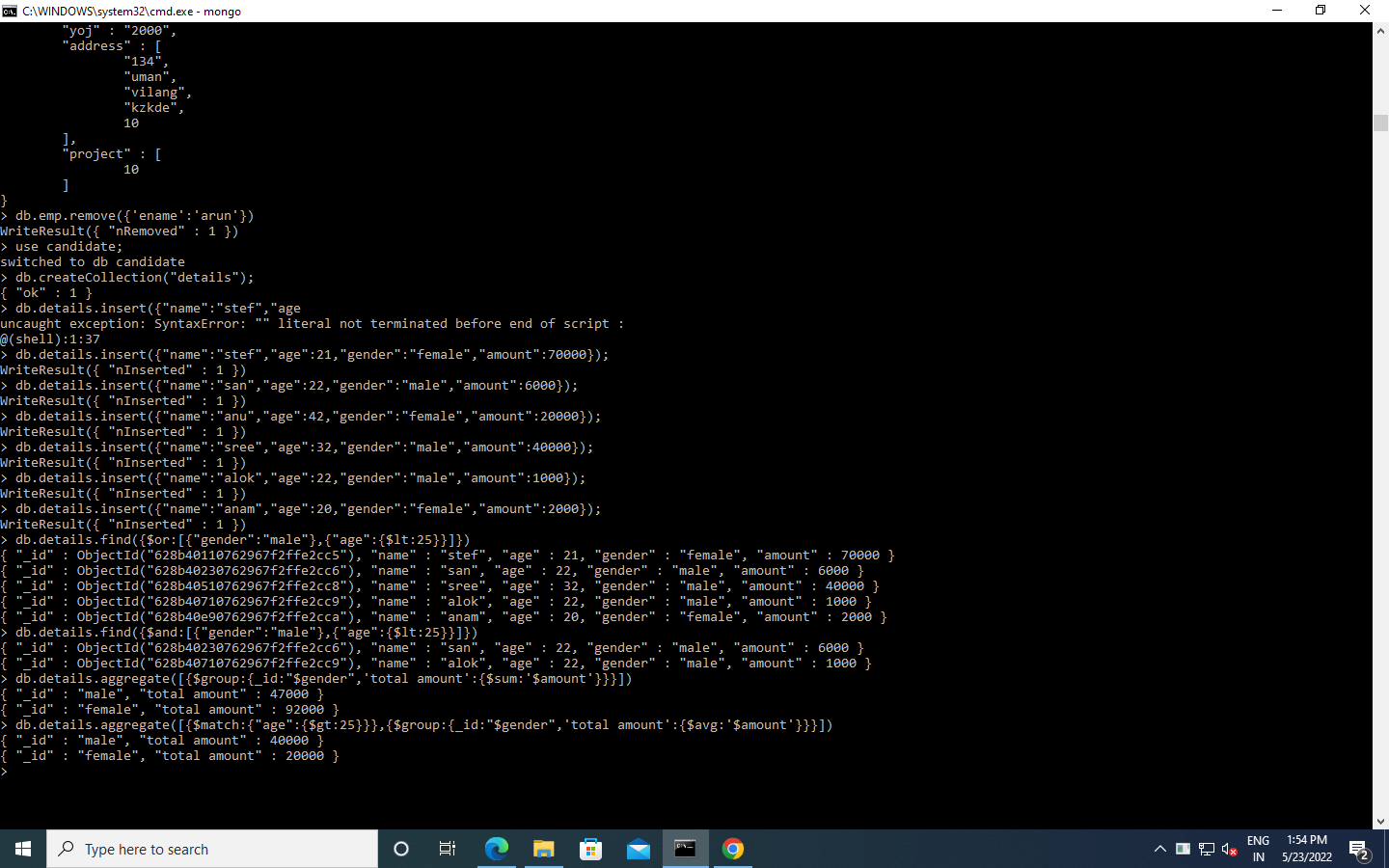
>db.details.insert({"name":"stef","age":21,"gender":"female","amount":7000});

> db.details.insert({"name":"san","age":22,"gender":"male","amount":6000});

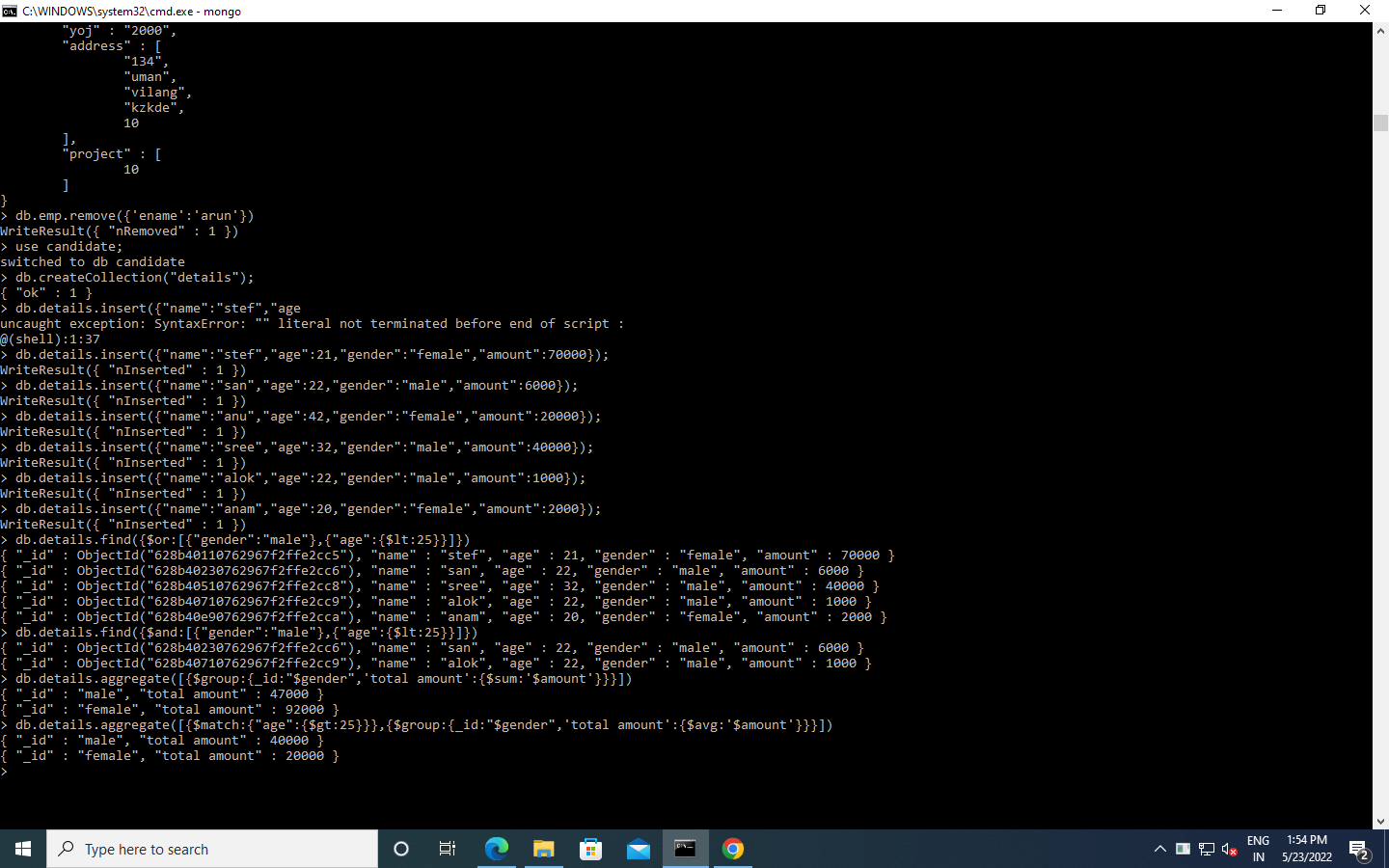
>db.details.insert({"name":"anu","age":42,"gender":"female","amount":20000})> db.details.insert({"name":"sree","age":32,"gender":"male","amount":40000});

>db.details.insert({"name":"anam","age":22,"gender":"female","amount":200);

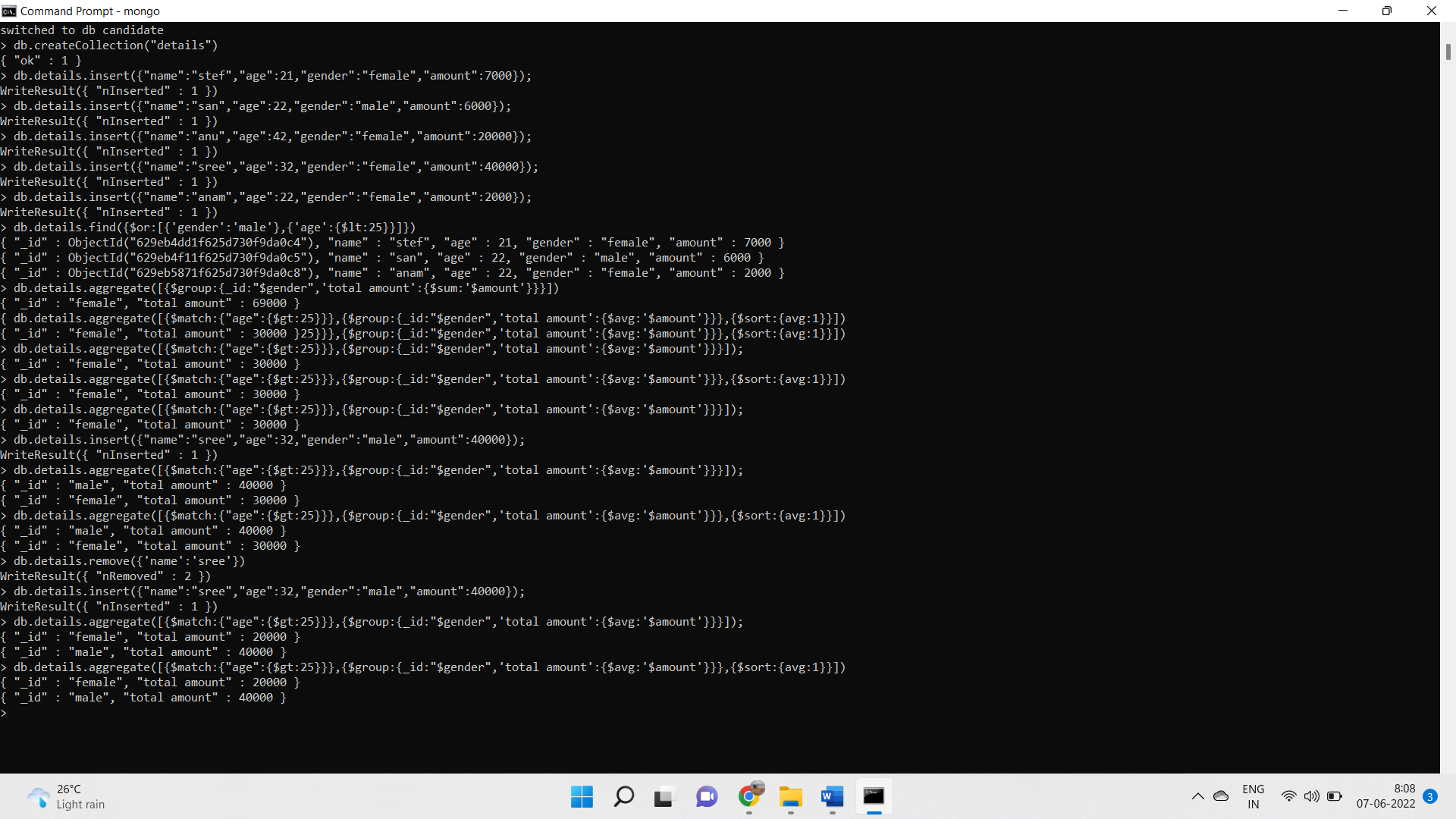
1. db.details.find({$or:[{'gender':'male'},{'age':{$lt:25}}]})



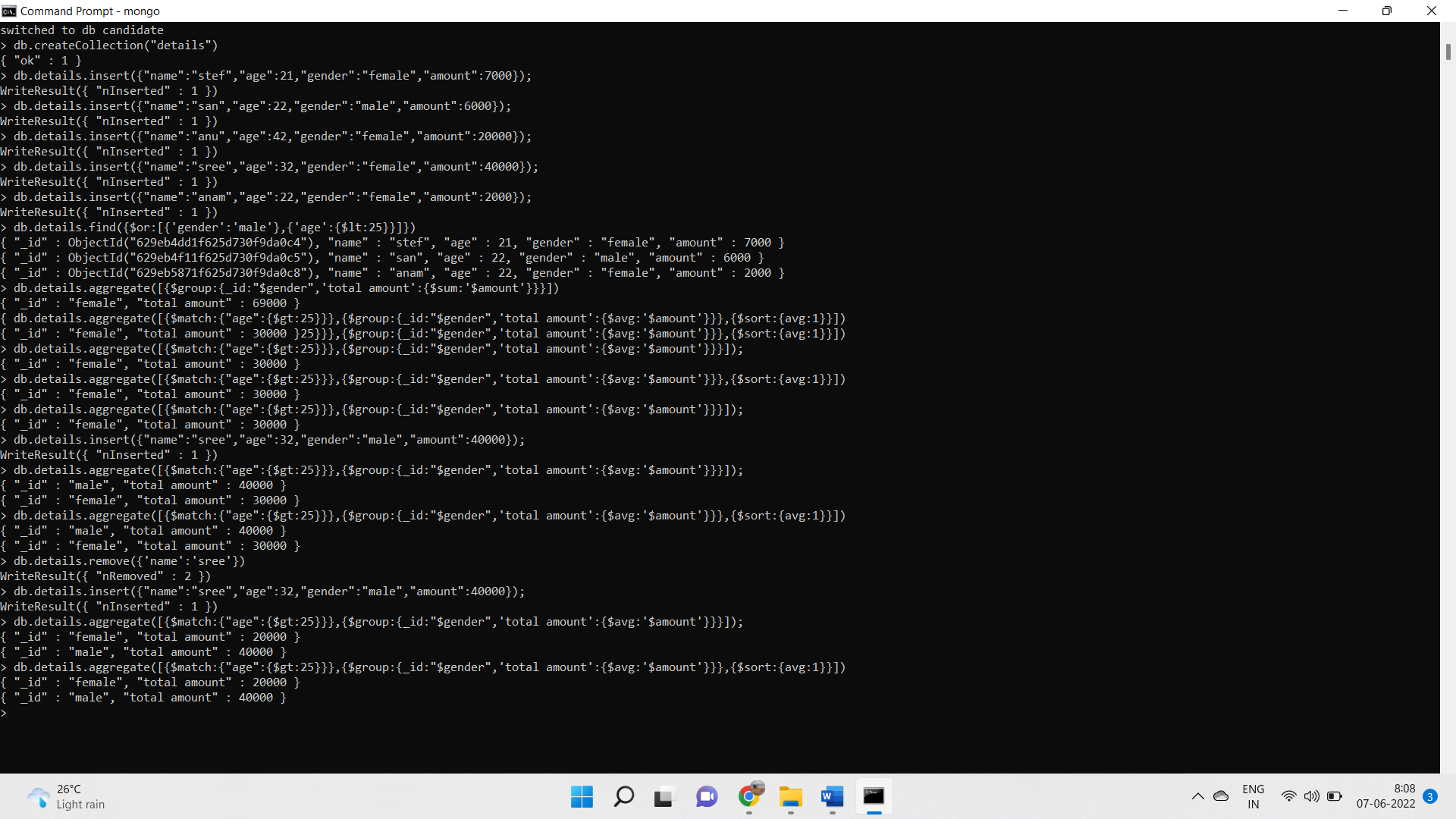
1. db.details.aggregate([{$group:{\_id:"$gender",'total amount':{$sum:'$amount'}}}])



1. db.details.aggregate([{$match:{"age":{$gt:25}}},{$group:{\_id:"$gender",'total amount':{$avg:'$amount'}}}])



4)db.details.aggregate([{$match:{"age":{$gt:25}}},{$group:{\_id:"$gender",'total amount':{$avg:'$amount'}}},{$sort:{avg:1}}])



**PROGRAM -6**

**AIM:** Create a database named college and then create a collection named studlist. Insert some values into it .

Write a MongoDB Query to:

1. Display name (both fname and lname) and mark of all female students in MCA department.

2. Display the details of student who secured highest mark in the course MCA

3. Display all male students who secured A+ grade.

4. Display the names of the top three students in Mechanical department.

5. Display the details of female students [fname,lname,grade,mark,contact] who achieved a mark more than 90.

6. Display the details of students who secured mark, more than 80 but less than 90.

7. Display the details of students whose name starts with ‘V’

8. Display all students from Kollam

9. Display all students who does not belong to neither Kollam nor Thiruvananthapuram

10. Display all female students who belong to either Kollam or Thiruvananthapuram

**CODE**:

use college;  
switched to db college;

> db.createCollection("studlist");

>db.studlist.insert({fname:"stefi",lname:"T",gender:"female",place:"kollam",dept:"mca",mark:"91",grade:"A",contact:"9946673258"});

>db.studlist.insert({fname:"sreelaya",lname:"vishal",gender:"female",place:"ernakulam",dept:"mca",mark:"89",grade:"A+",contact:"9867453622"});

>db.studlist.insert({fname:"sandhya",lname:"sanu",gender:"male",place:"trivandrum",dept:"mechanical",mark:"40",grade:"A",contact:"9756348950"});

>db.studlist.insert({fname:"sanoop",lname:"s",gender:"male",place:"kollam",dept:"mechanical",mark:"95",grade:"A+",contact:"954128740"});

>db.studlist.insert({fname:"anu",lname:"ashok",gender:"female",place:"idukki",dept:"mechanical",mark:"85",grade:"A",contact:"9784902124"});

>db.studlist.insert({fname:"vismaya",lname:"v",gender:"female",place:"thrissur",dept:"mca",mark:"99",grade:"A+",contact:"9865124785"});

>db.studlist.insert({fname:"vinaya",lname:"v",gender:"female",place:"trivandrum",dept:"mechanical",mark:"98",grade:"A+",contact:"9854874545"});

1. > db.studlist.find({gender:'female',dept:'mca'},{fname:1,lname:1})
2. > db.studlist.find().sort({"mark":-1}).limit(1)
3. >db.studlist.find({gender:'male',grade:'A+'},{fname:1,lname:1,gender:1,dept:1,loc:1,grade:1,mark:1})
4. > db.studlist.find({dept:"mechanical"}).sort({mark:-1}).limit(3)
5. >db.studlist.find({mark:{$gt:90},gender:'female'},{fname:1,lname:1,mark:1,contact:1,grade:1})
6. >db.studlist.find({$and:[{mark:{$gt:80}},{mark:{$lt:90}}]},{fname:1,lname:1,mark:1,contact:1,grade:1})
7. > db.studlist.find({fname:{$regex:'^v'}},{})
8. > db.studlist.find({place:"kollam"},{})
9. > db.studlist.find({$nor:[{place:"kollam"},{loc:"trivandrum"}]},{})
10. >db.studlist.find({$or:[{place:"kollam"},{loc:"trivandrum"}],gender:"female"},{})

**OUTPUT**:

