PROIECT ELEMENTE AVANSATE DE BAZE DE DATE

-MongoDB-

NUME STUDENT: Dănilă Ana-Maria

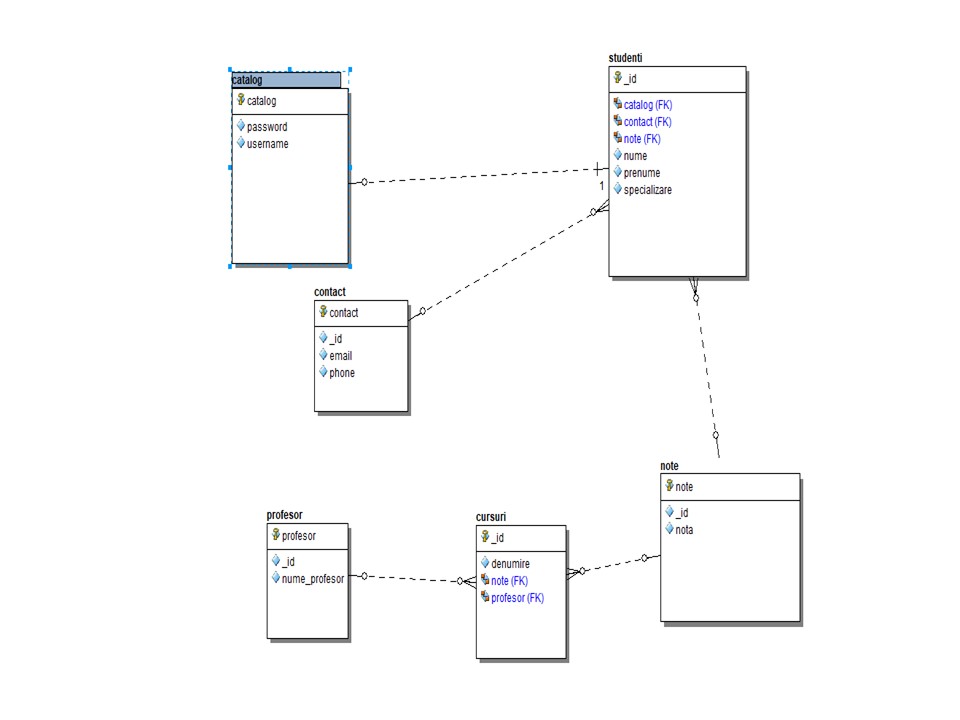
GRUPA:1

SEMIGRUPA: 1

CATALOG CURSURI OPȚIONALE FSEGA

SCENARIU

.



Schema bazei de date

**COLECȚII CREATE**: STUDENȚI, CURSURI

Colecții și documente

În colecția STUDENȚI avem 5 documente(înregistrări). Fiecare document din această colecție are următorele attribute de tip cheie:

**\_id**🡪modul de identificare al fiecărui student(îl setăm explicit, pentru a ne putea referi mai ușor la studenți atunci când vo crea colecția CURSURI

**Nume,Prenume,Specializare**

De asemenea am ales să includ în colecția STUDENȚI și **documente imbricate**:

-**contact**(fiecare student are un singur **număr de telefon** și un singur **emai**l trecut în baza de date, iar fiecare email sau număr de telefon corespunde unui singur student->**relație one to one**)

Am ales să nu fac o colecție separată cu contactele studenților deoarece va fi mai ușor să facem interogări care să ne extragă date din aceeași colecție decât din colecții diferite.

**-catalog(** fiecare student are propriul său catalog deoarecefiecare dintre ei primește un username și o parolă diferită, însă catalogul conține username-ul și parola fiecăruia, deci este o **relație one to many)**

-Având în vedere că fiecare student are un username și o parolă și între catalog și studenți este o relație one to many este logic ca datele fiecărui student să fie în aceeași colecție cu celelalte informații personale. În același timp interogarea unei singure colecții se realizează mai ușor decât atunci când datele sunt în colecții diferite.(Să presupunem că în viitor conducerea are nevoie de numele și de numărul de telefon al fiecărui student-pentru a le obține va interoga o singură colecție).

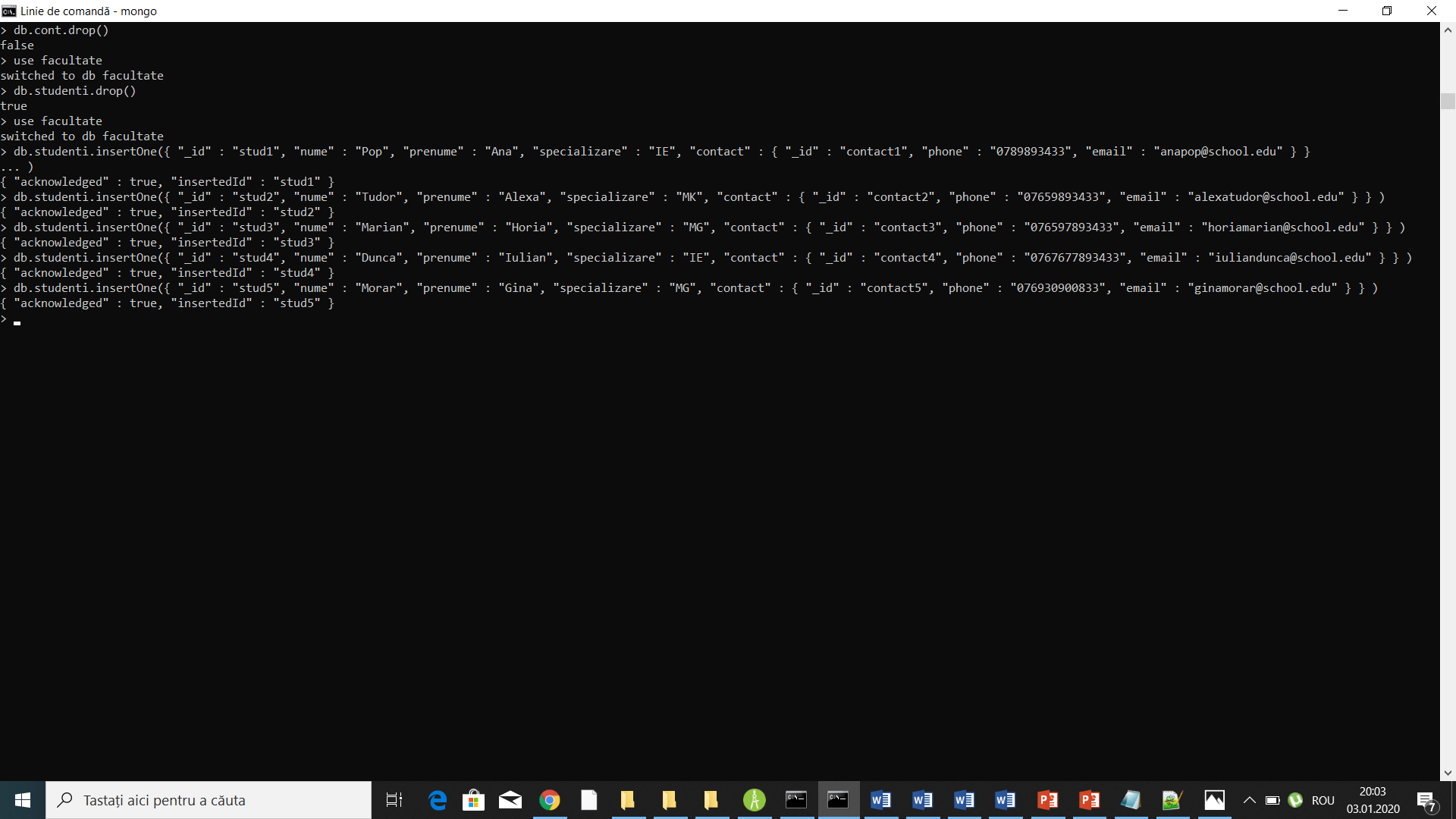
În această colecție avem și o listă de obiecte numită **note** ce conține id-ul cursurilor la care s-a înscris studentul. Am ales să fac o listă pentru a putea adăuga cursuri noi sau modifica cursurile mult mai ușor.

În colecția CURSURI avem **denumirea** fiecărui curs, **id-ul** cursului, un subdocument **profesor** ce conține cheile **id** și **nume** și o listă de obiecte care are ca și chei **id-ul** studentului și **nota** obținută de acel student la acel curs.

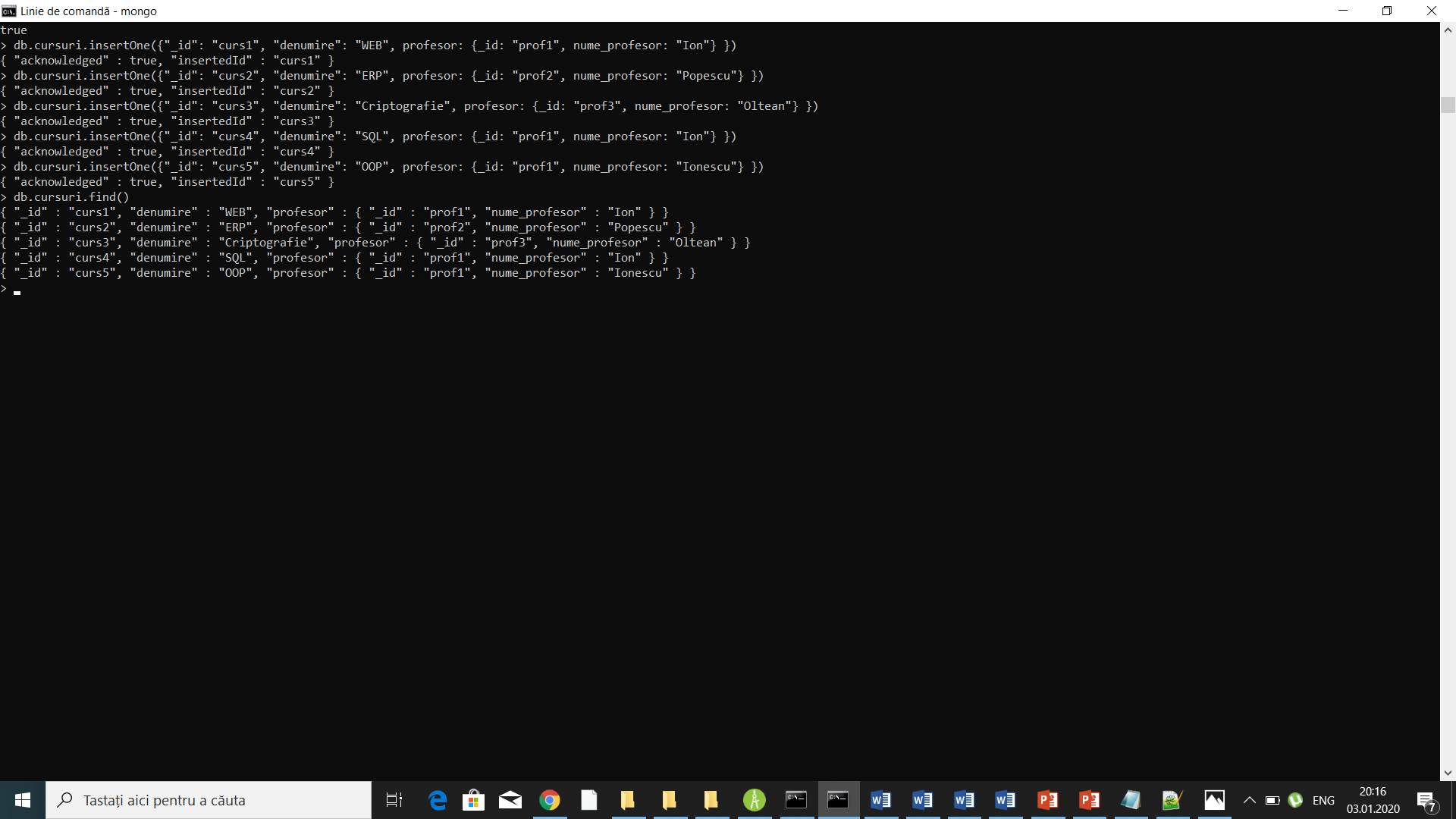
-Am ales să creez un subdocument profesor și nu o colecție separate deoarece între profesor și cursuri este o **relație one to many-**un profesor poate preda mai multe cursuri, dar un curs poate fi predat doar de un profesor

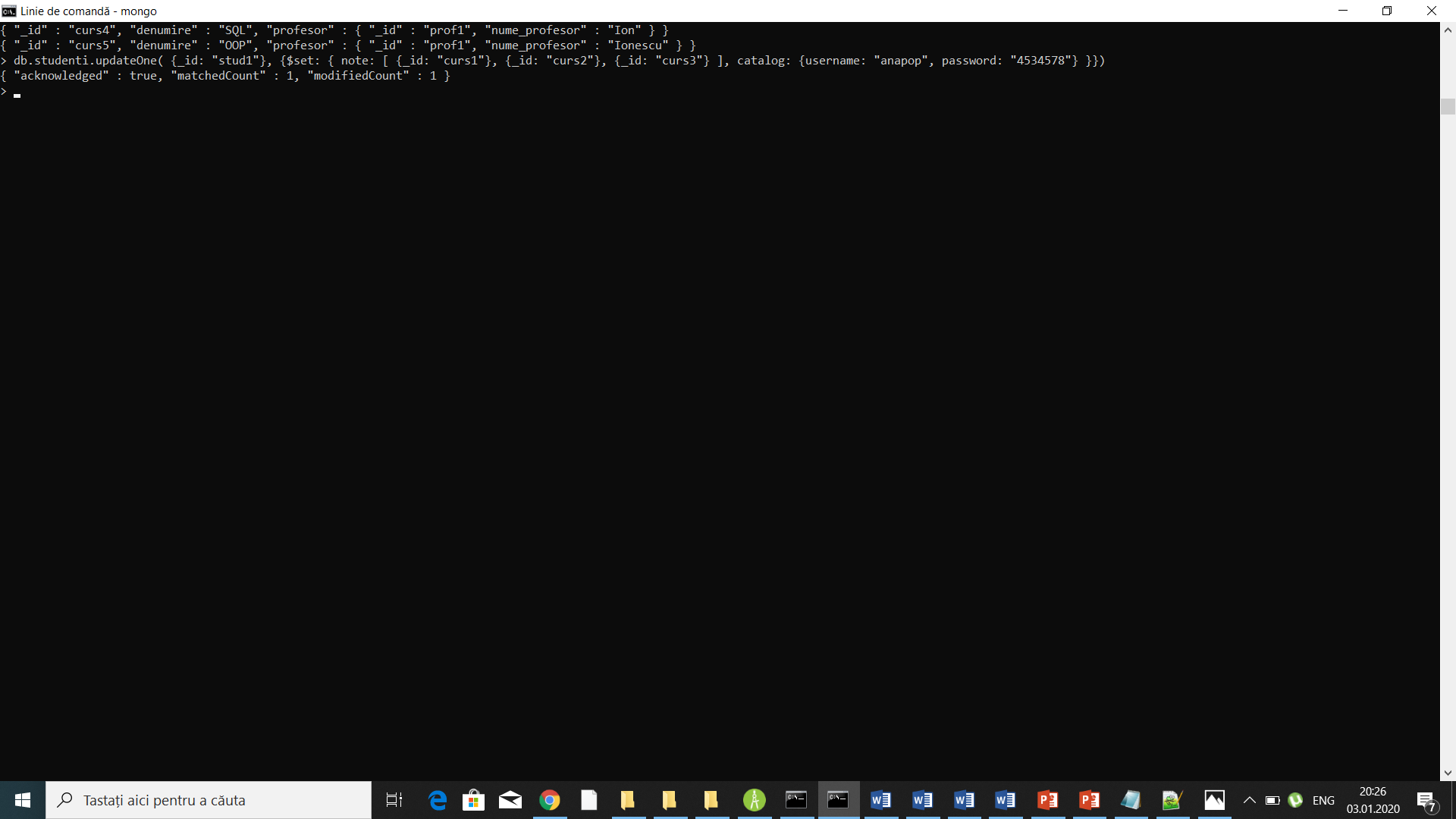
-Am ales să creez subdocumentul note în colecția CURSURI deoarece la fiecare curs mai mulți studenți obțin mai multe note, în timp ce la același curs un student nu poate obține decât o notă-**relație one to many**

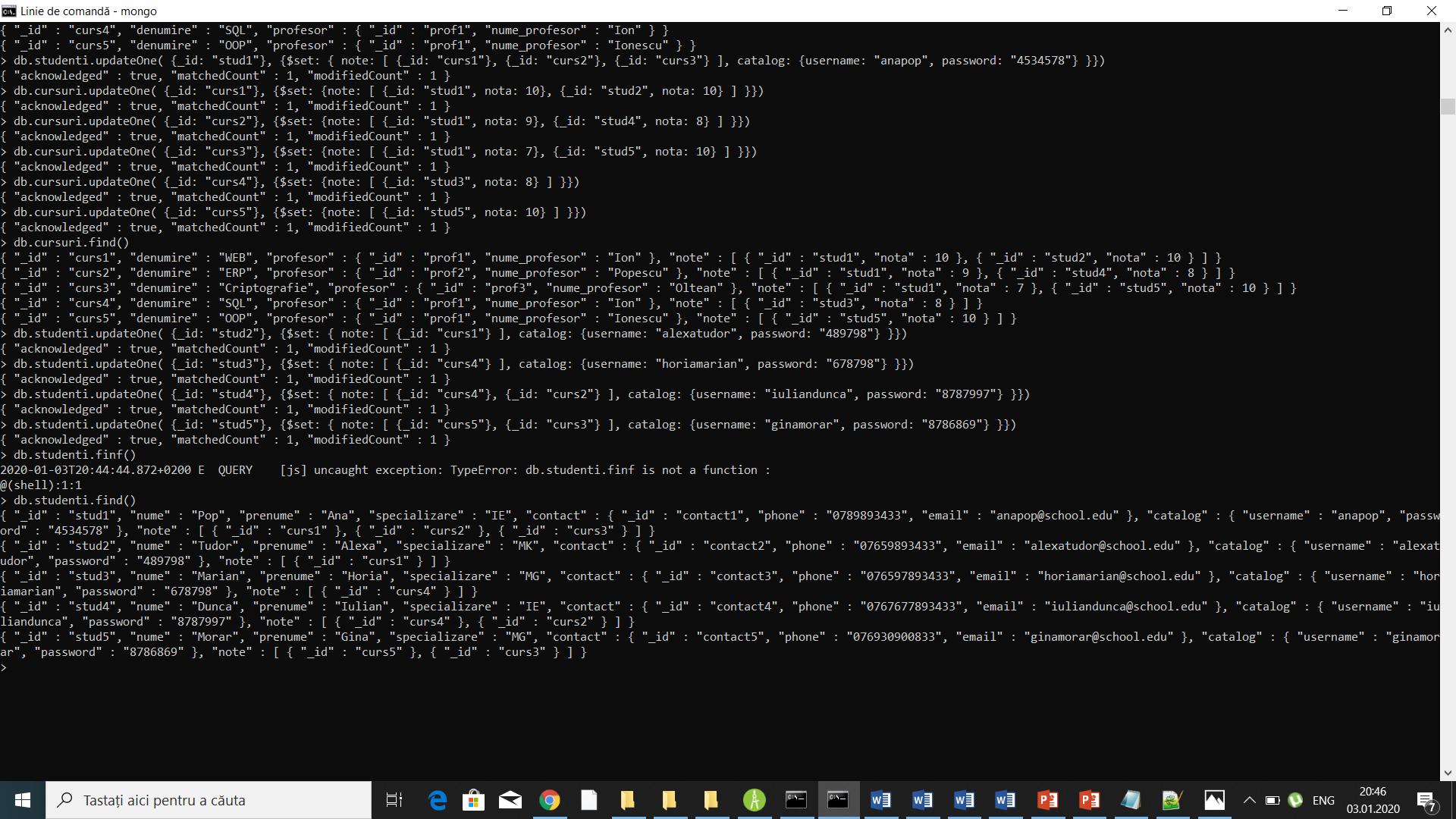
**Implementarea bazei de date folosind clientul mongo.exe**

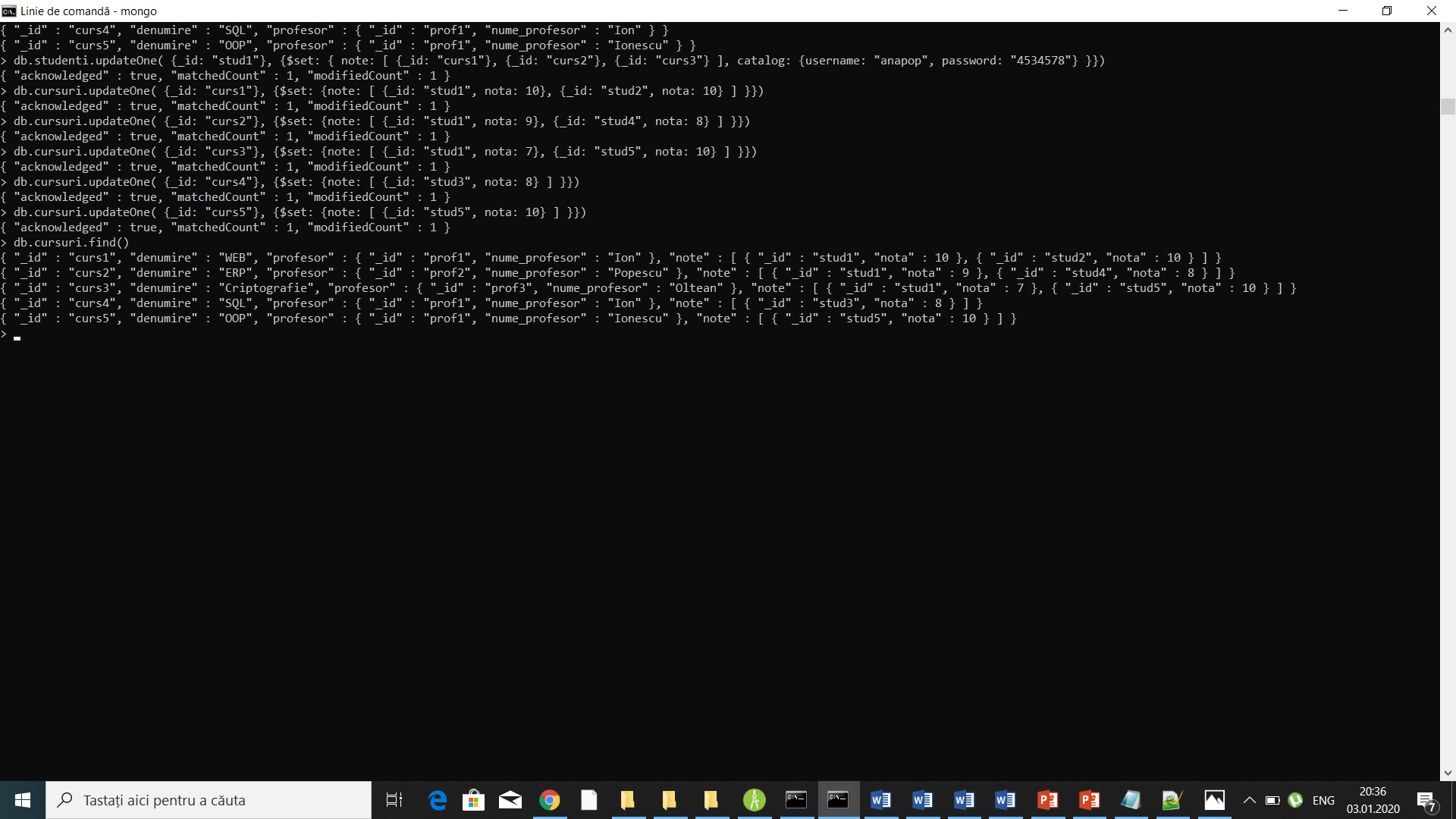


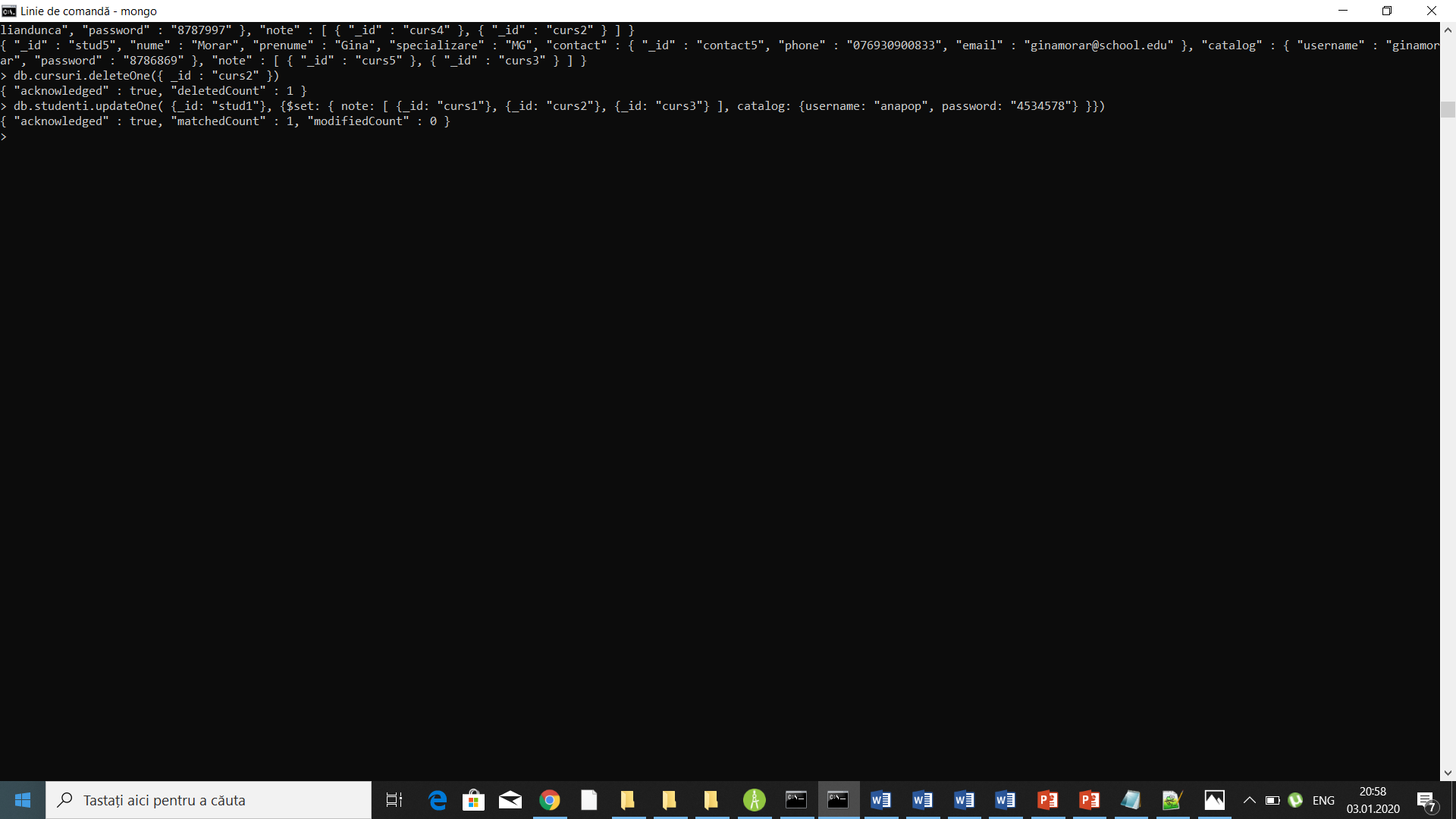
STUDENȚI

CURSURI

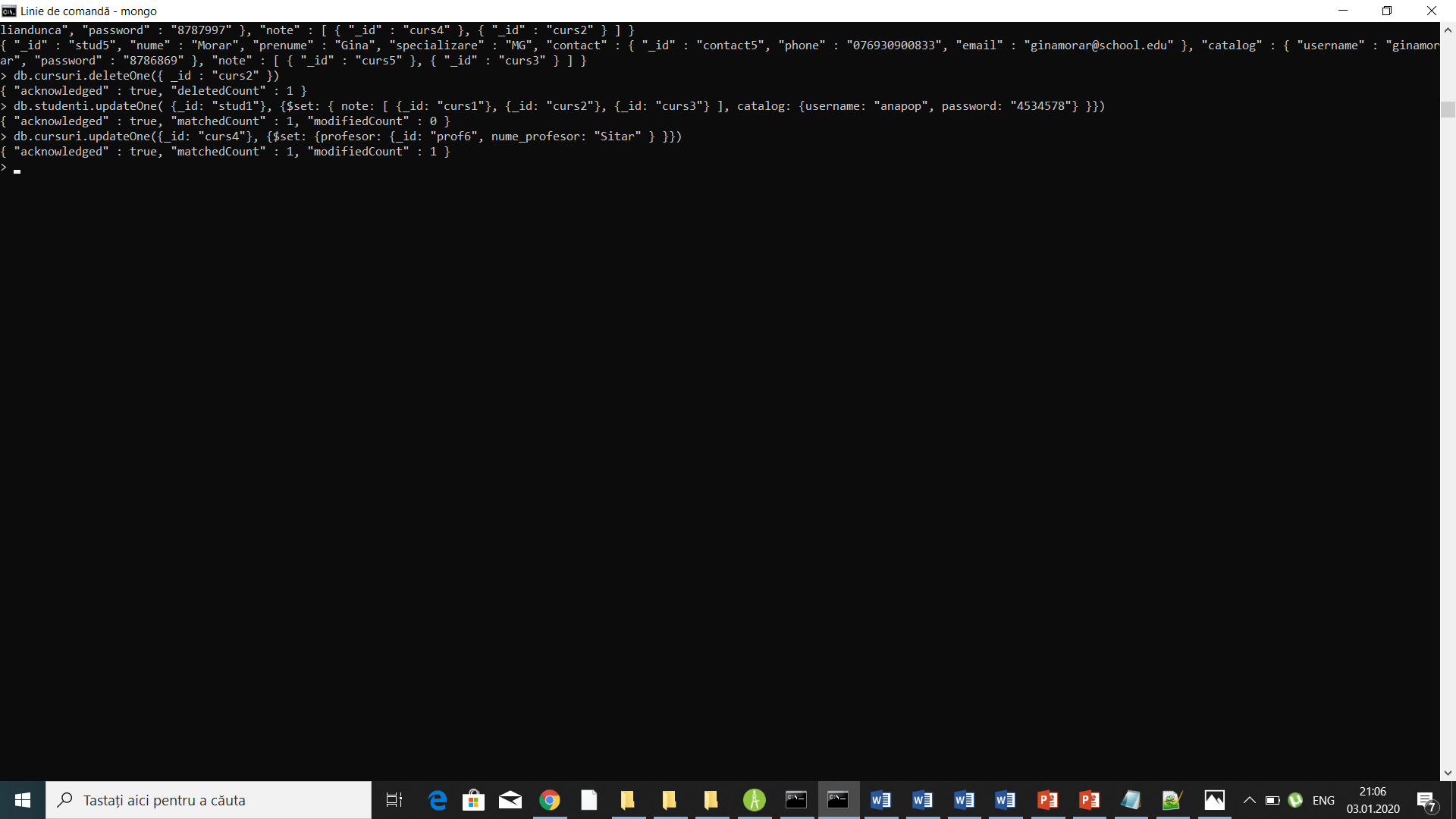
Facem update și asignăm fiecărui student un curs creat și un username și o parolă



Facem update la CURSURI și asignăm fiecărui curs studenții creați și notele acestora.

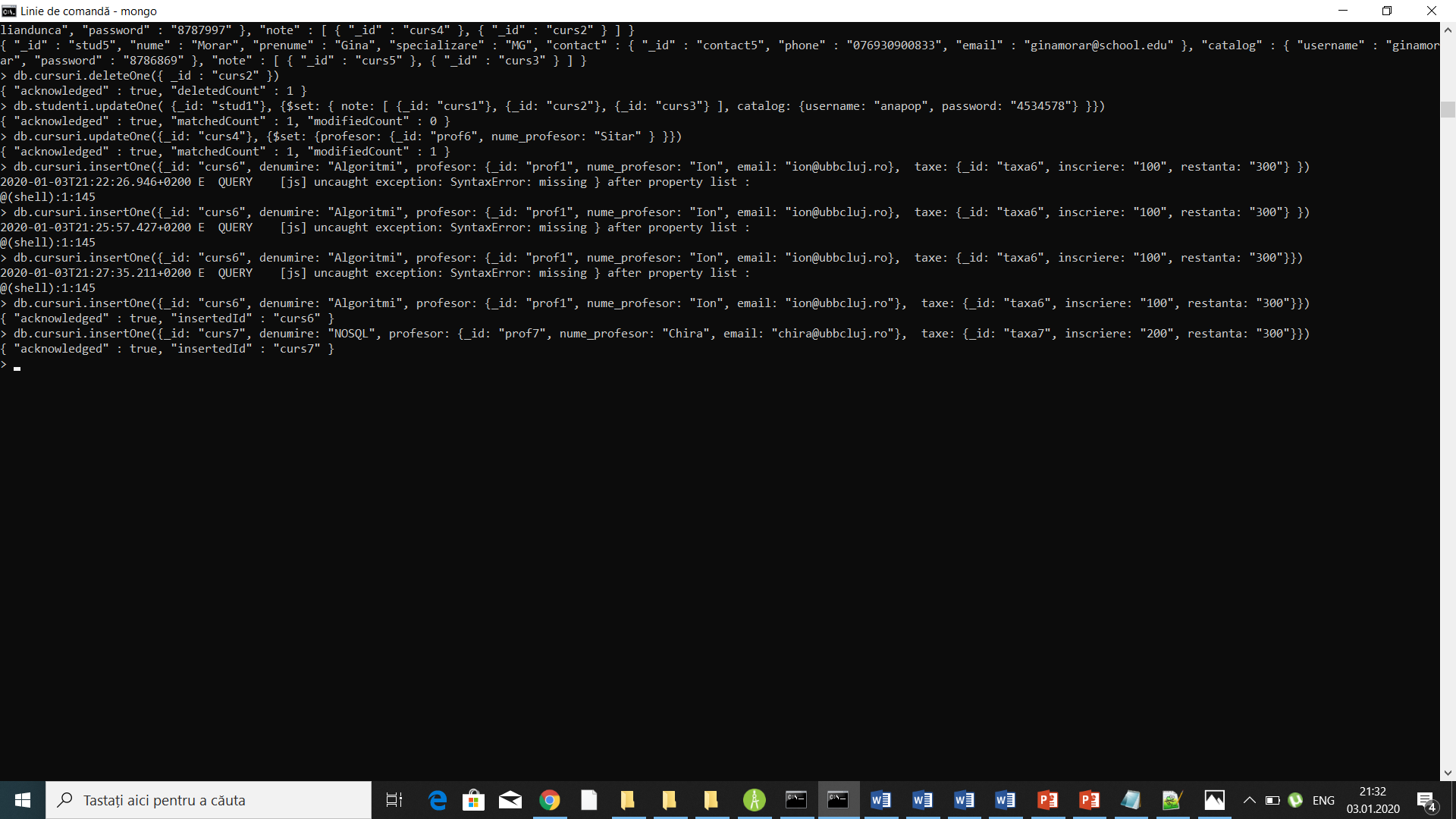
ștergem cursul2

db.cursuri.deleteOne({ \_id : "curs2" })



UPDATE la cursul 4

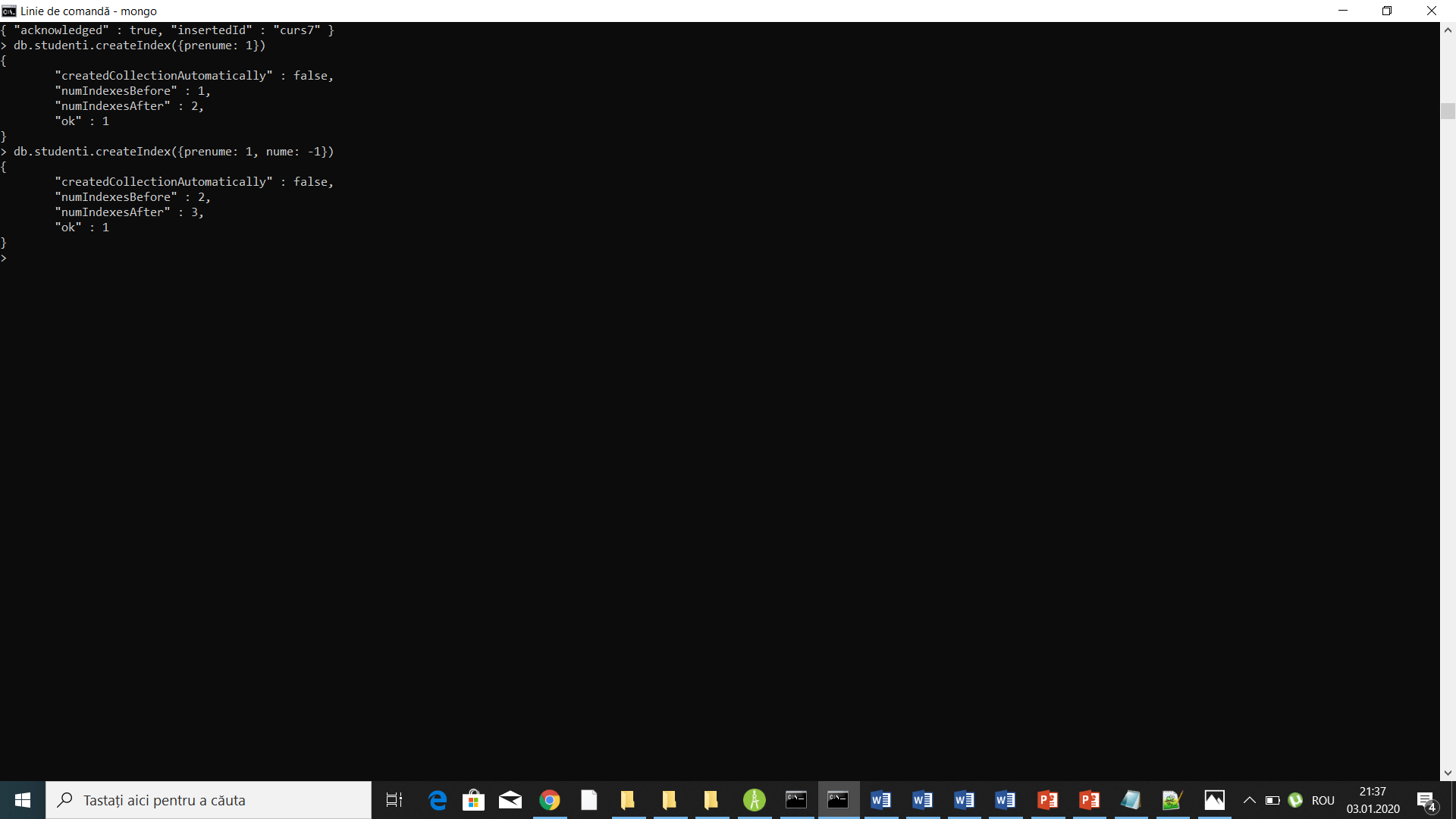
db.cursuri.updateOne({\_id: "curs4"}, {$set: {profesor: {\_id: "prof6", nume\_profesor: "Sitar" } }})

INSERĂM CURSURI NOI

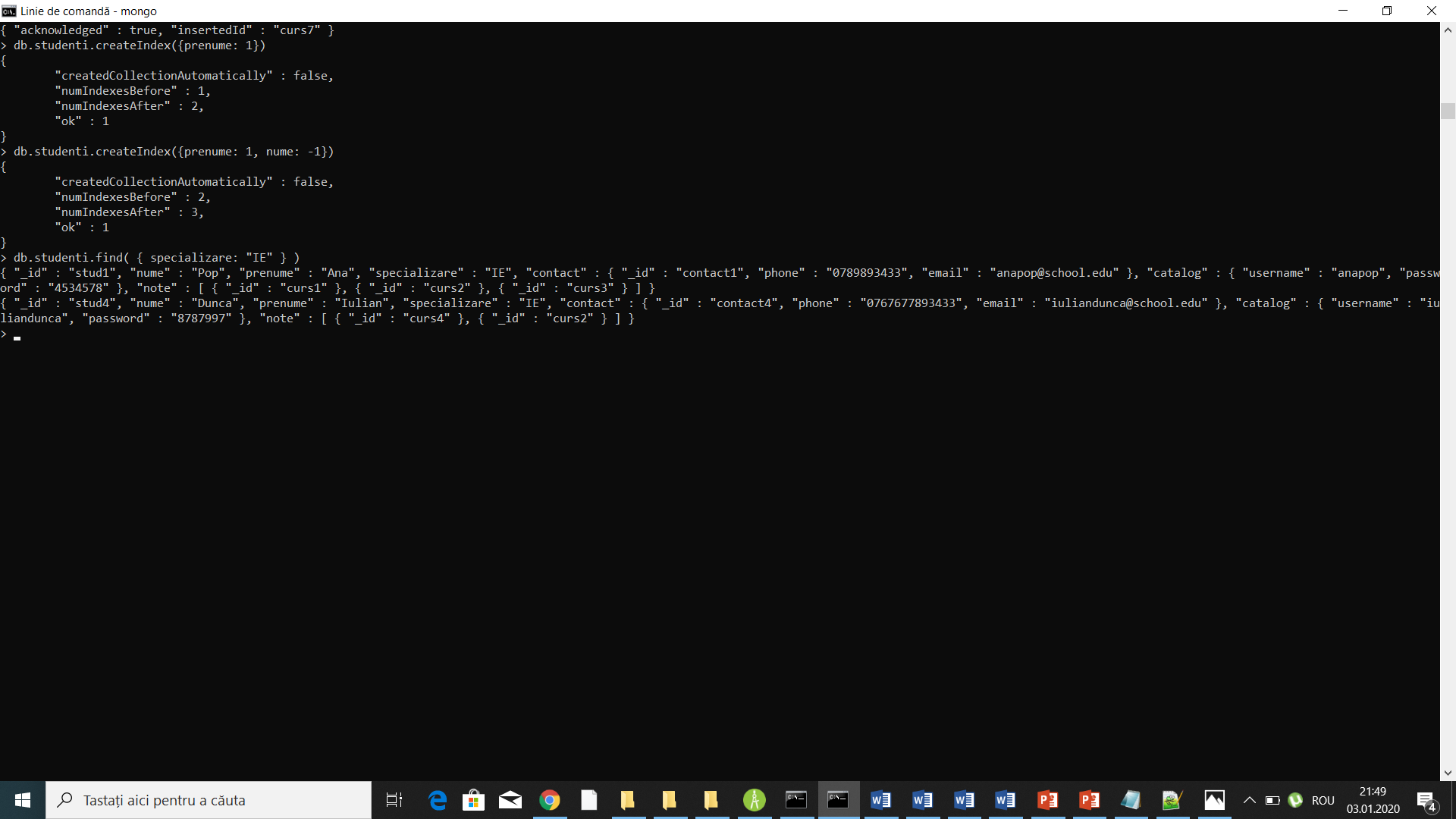
db.cursuri.insertOne({\_id: "curs6", denumire: "Algoritmi", profesor: {\_id: "prof1", nume\_profesor: "Ion", email: "ion@ubbcluj.ro"}, taxe: {\_id: "taxa6", inscriere: "100", restanta: "300"}})

db.cursuri.insertOne({\_id: "curs7", denumire: "NOSQL", profesor: {\_id: "prof7", nume\_profesor: "Chira", email: "chira@ubbcluj.ro"}, taxe: {\_id: "taxa7", inscriere: "200", restanta: "300"}})

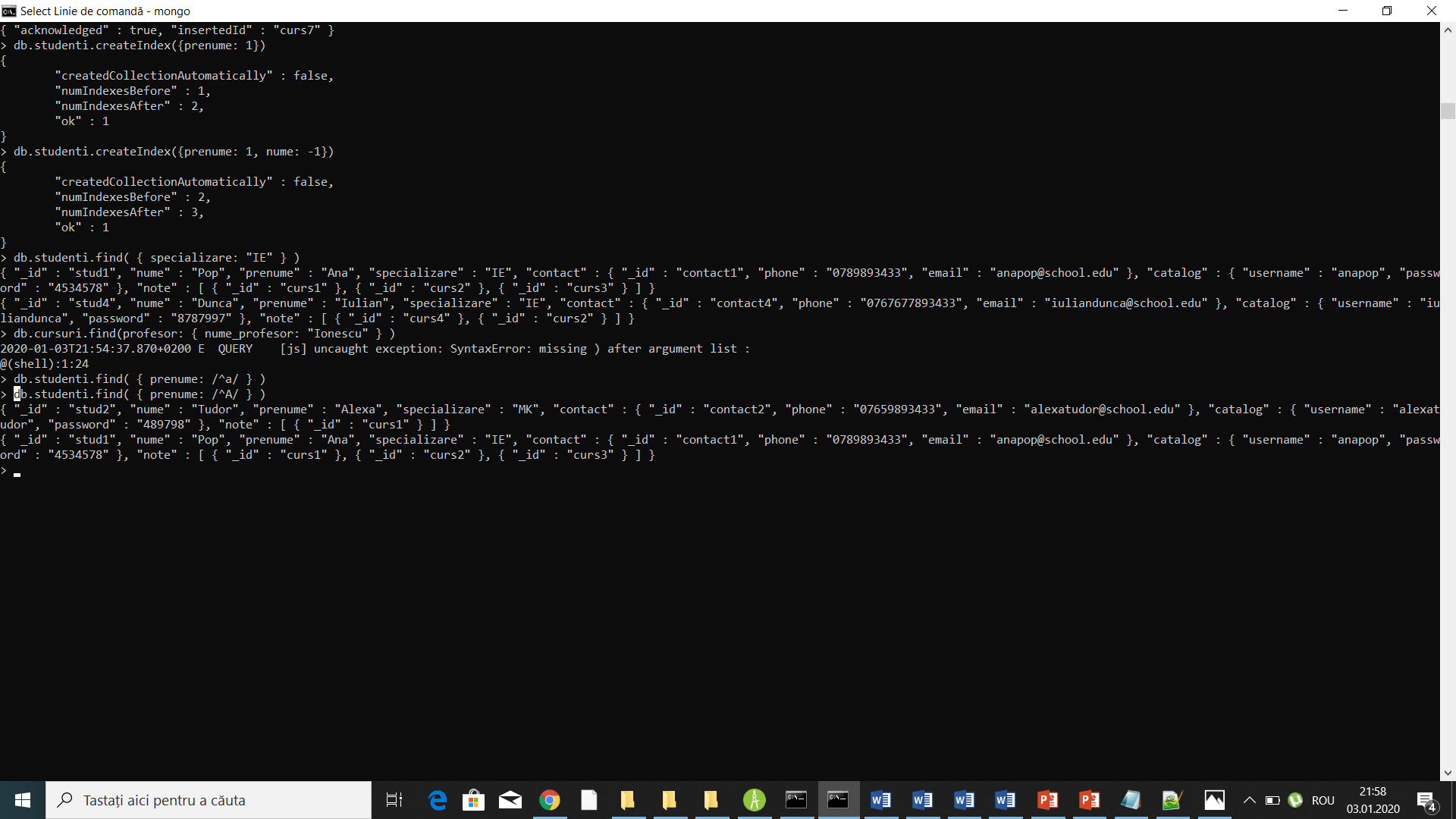
Aceste documente vor fi updatate după ce studenții se vor înscrie la cursuri.

INDEX SIMPLU ȘI COMPUS

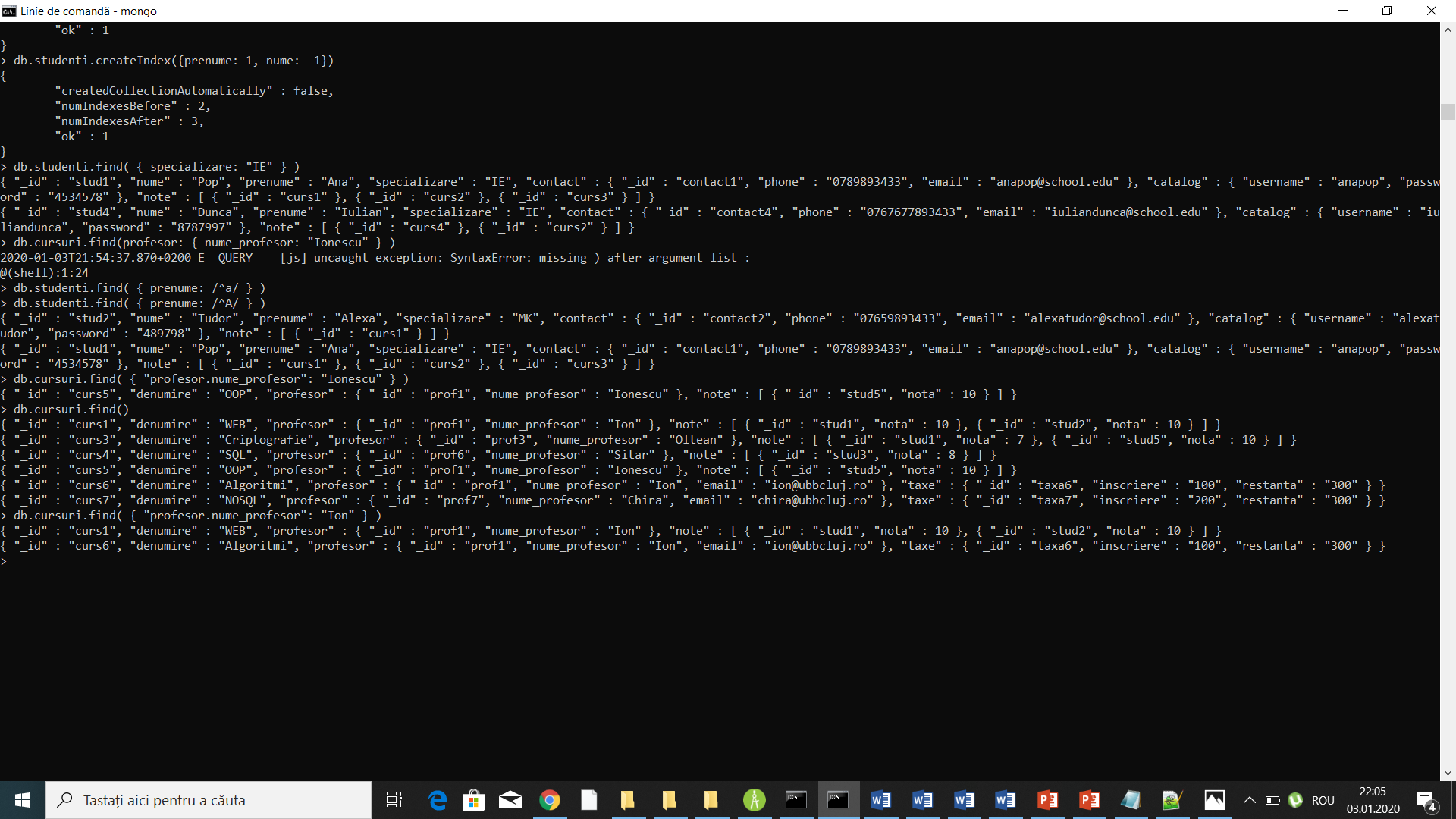
INTEROGĂRI

**1.Să se afișeze studenții de la specializare IE.**

db.studenti.find( { specializare: "IE" } )

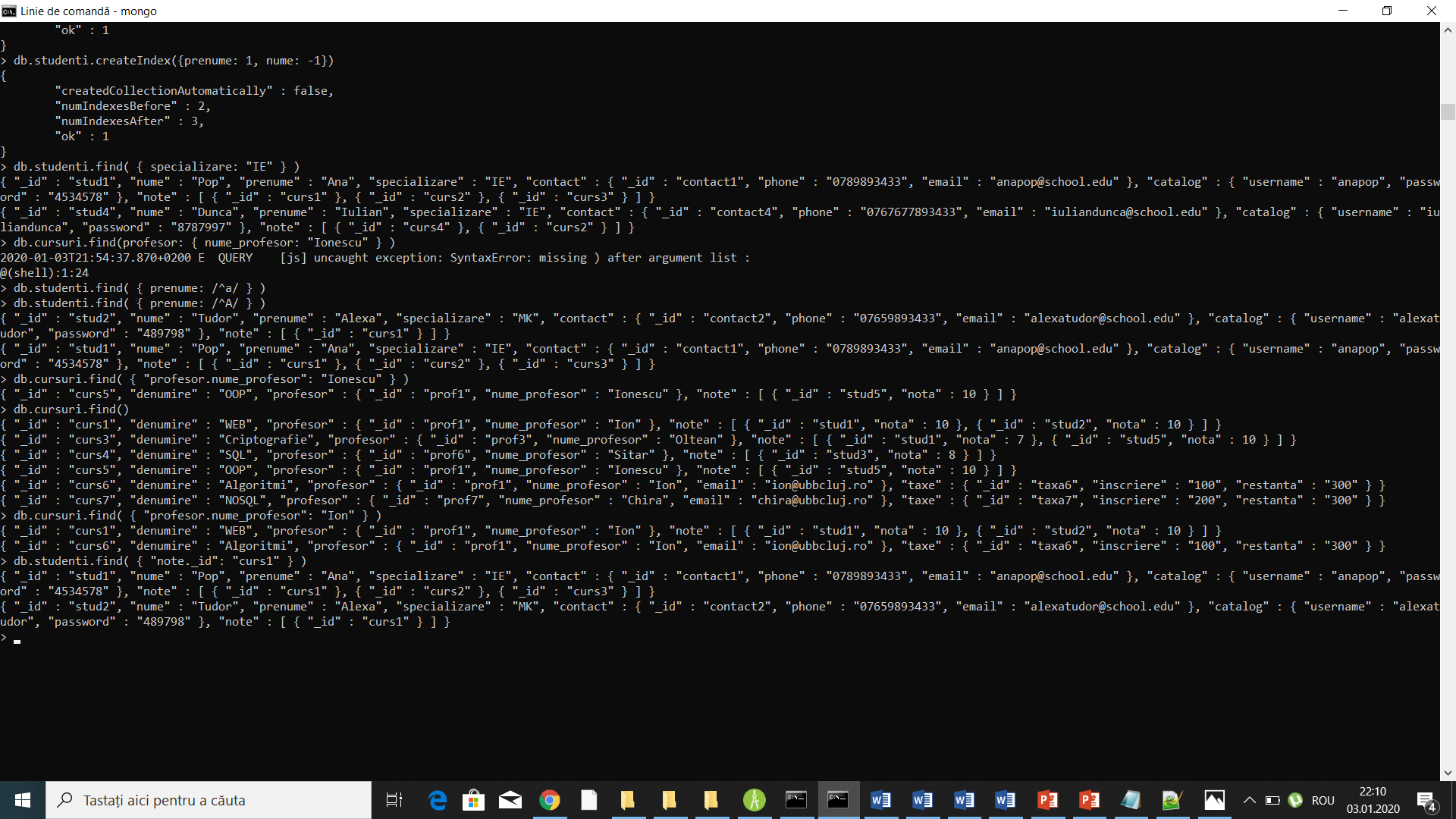
2.**Să se afișeze studenții a căror prenume începe cu A.**

db.studenti.find( { prenume: /^A/ } )

3**.Să se afișeze toate cursurile predate de profesorul Ion.**

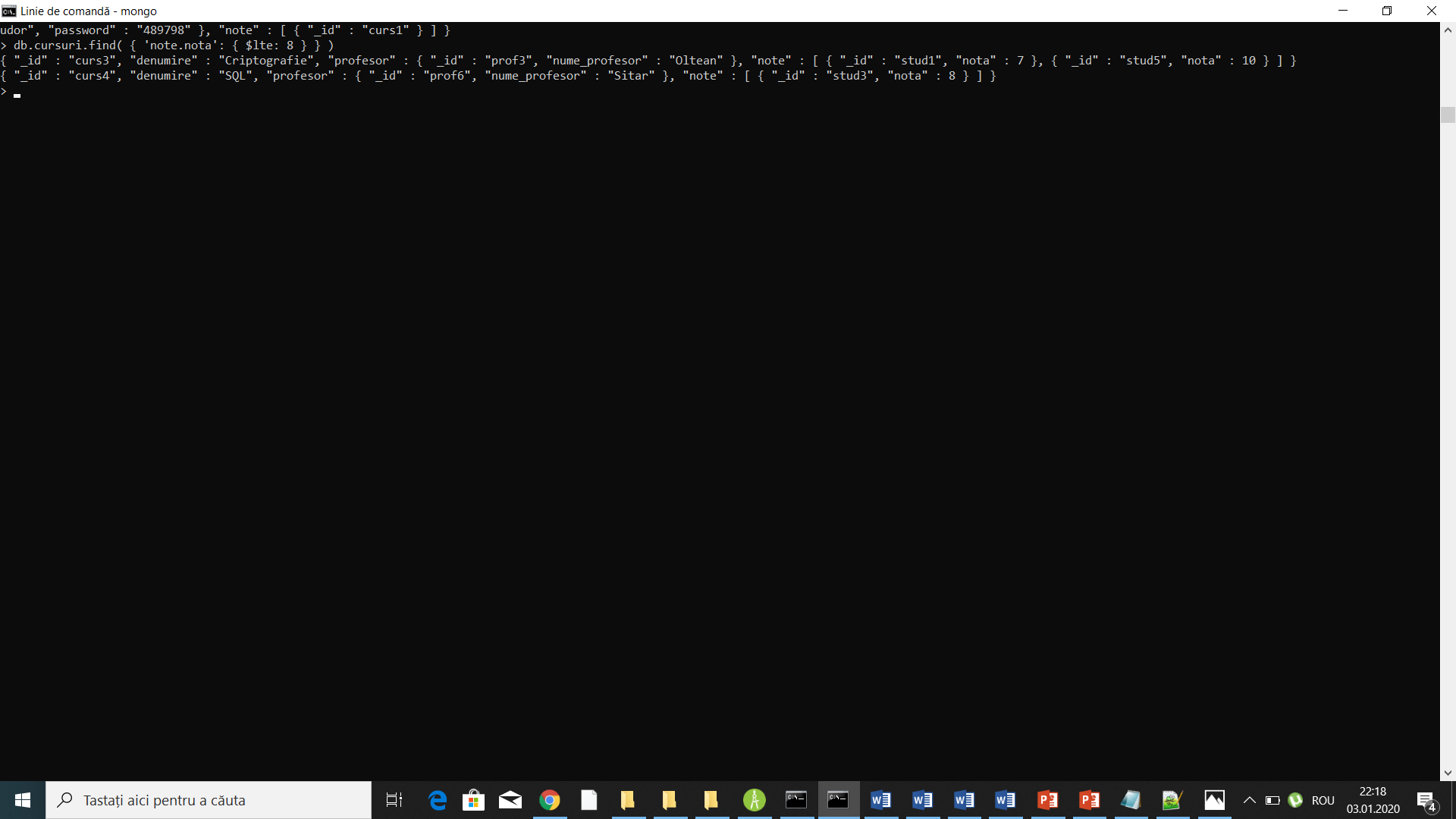
db.cursuri.find( { "profesor.nume\_profesor": "Ion" } )

**4.Să se afișeze toți studenții care s-au înscris la cursul cu id-ul curs1.**



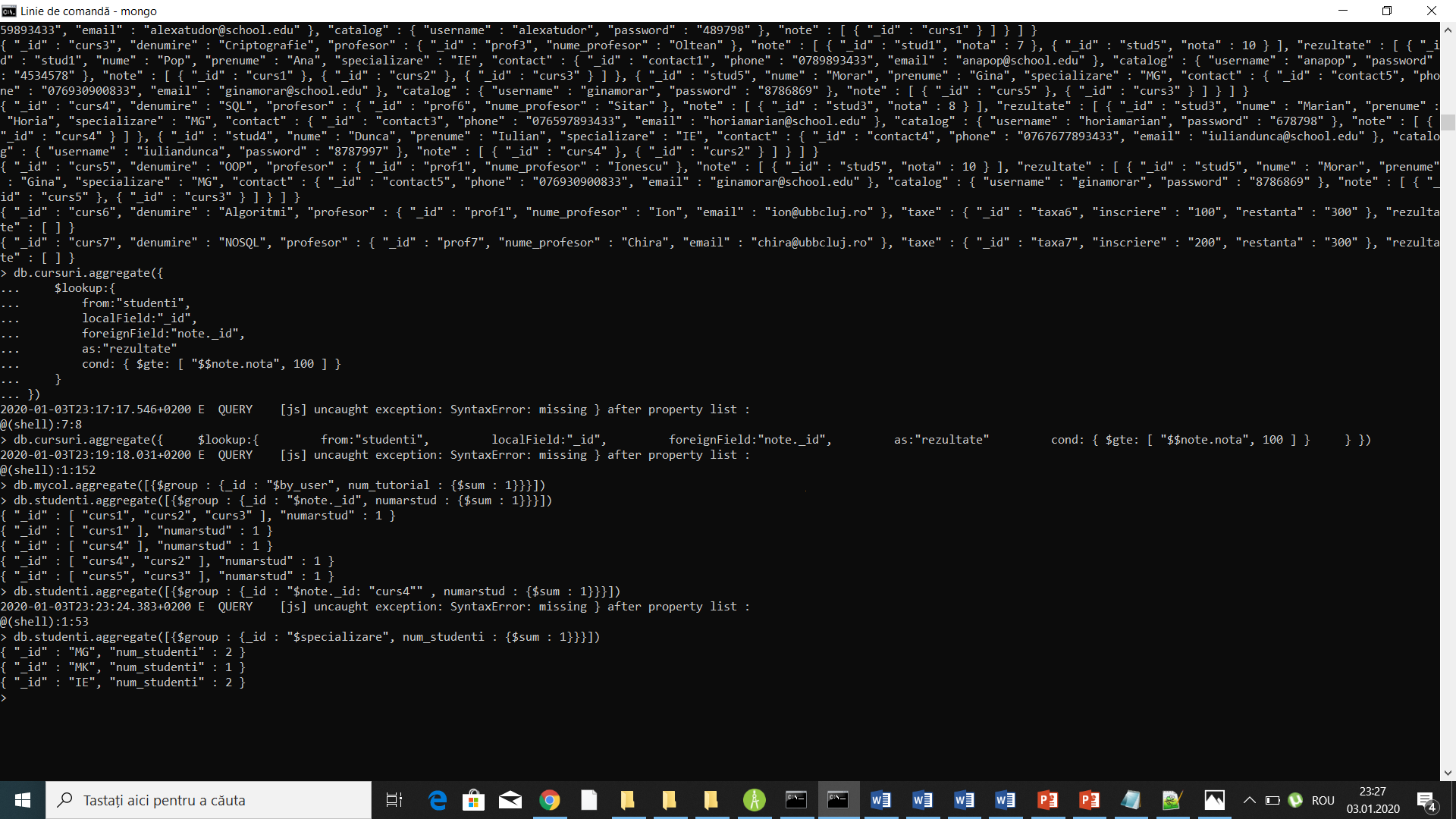
db.studenti.find( { "note.\_id": "curs1" } )

5.**Să se afișeze toate cursurile la care cel puțin un student a obținut o notă mai mică de 8**



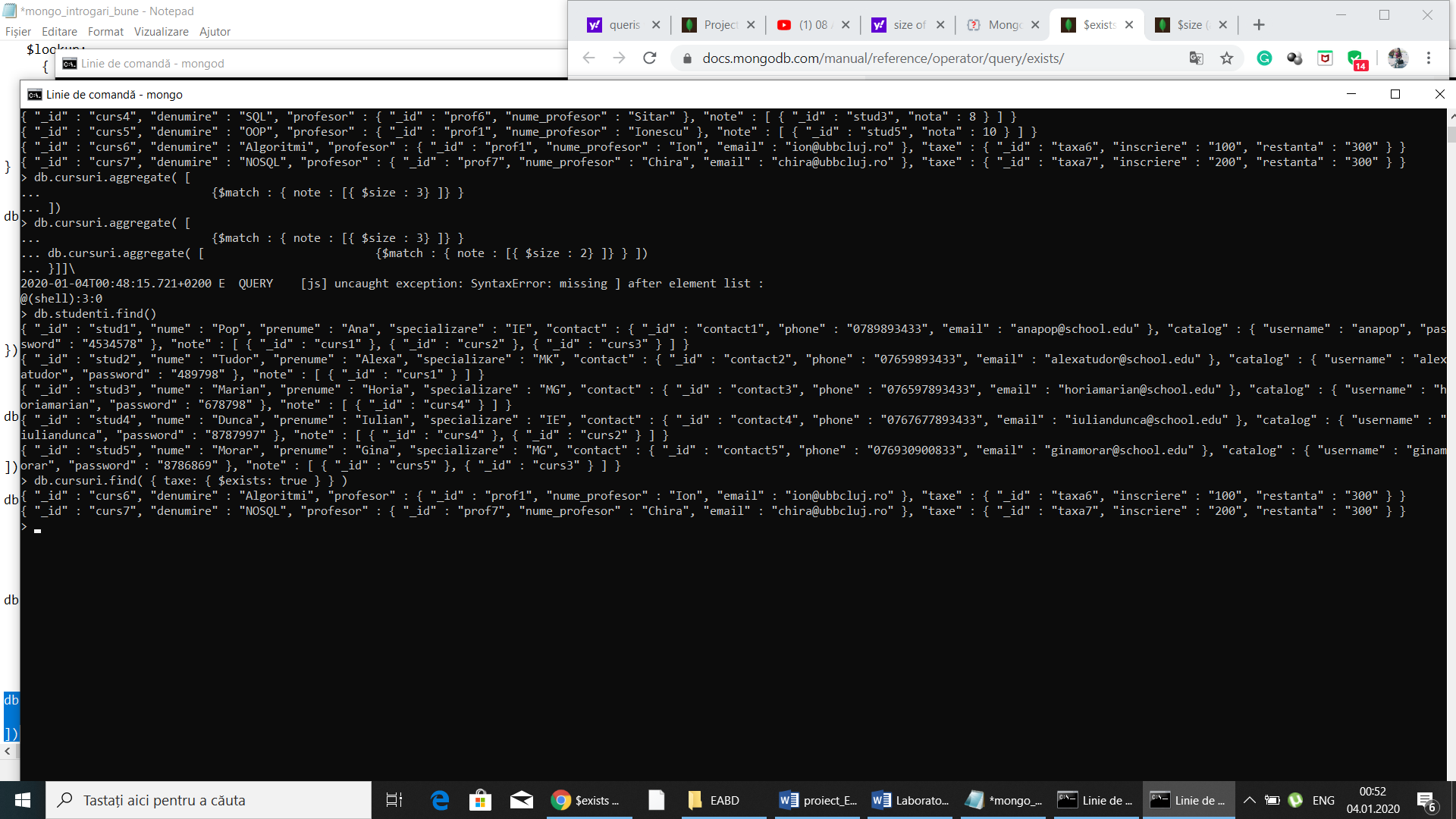
db.cursuri.find( { 'note.nota': { $lte: 8 } } )

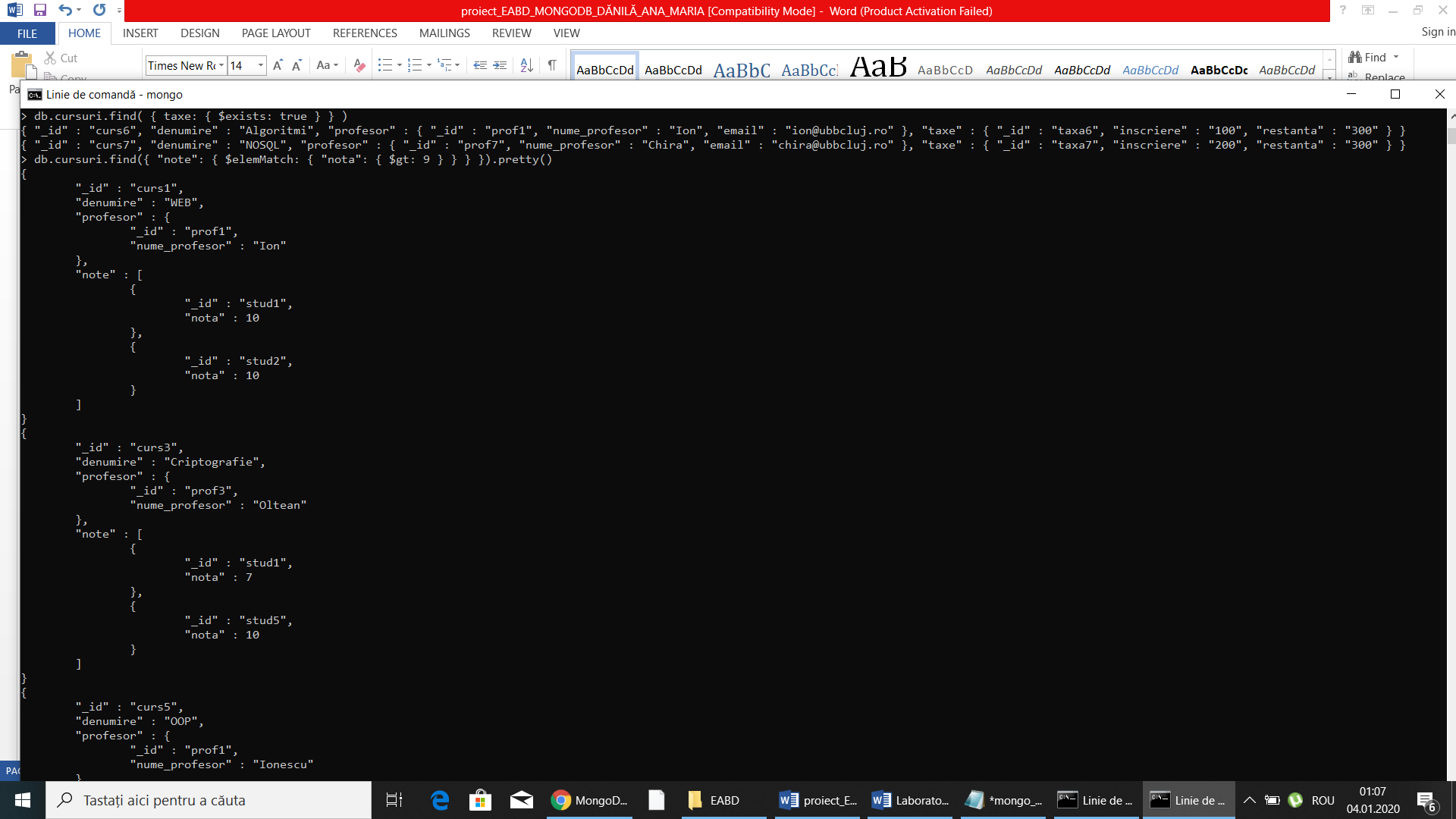
6**.Să se afișeze câți studenți sunt la fiecare specializare**



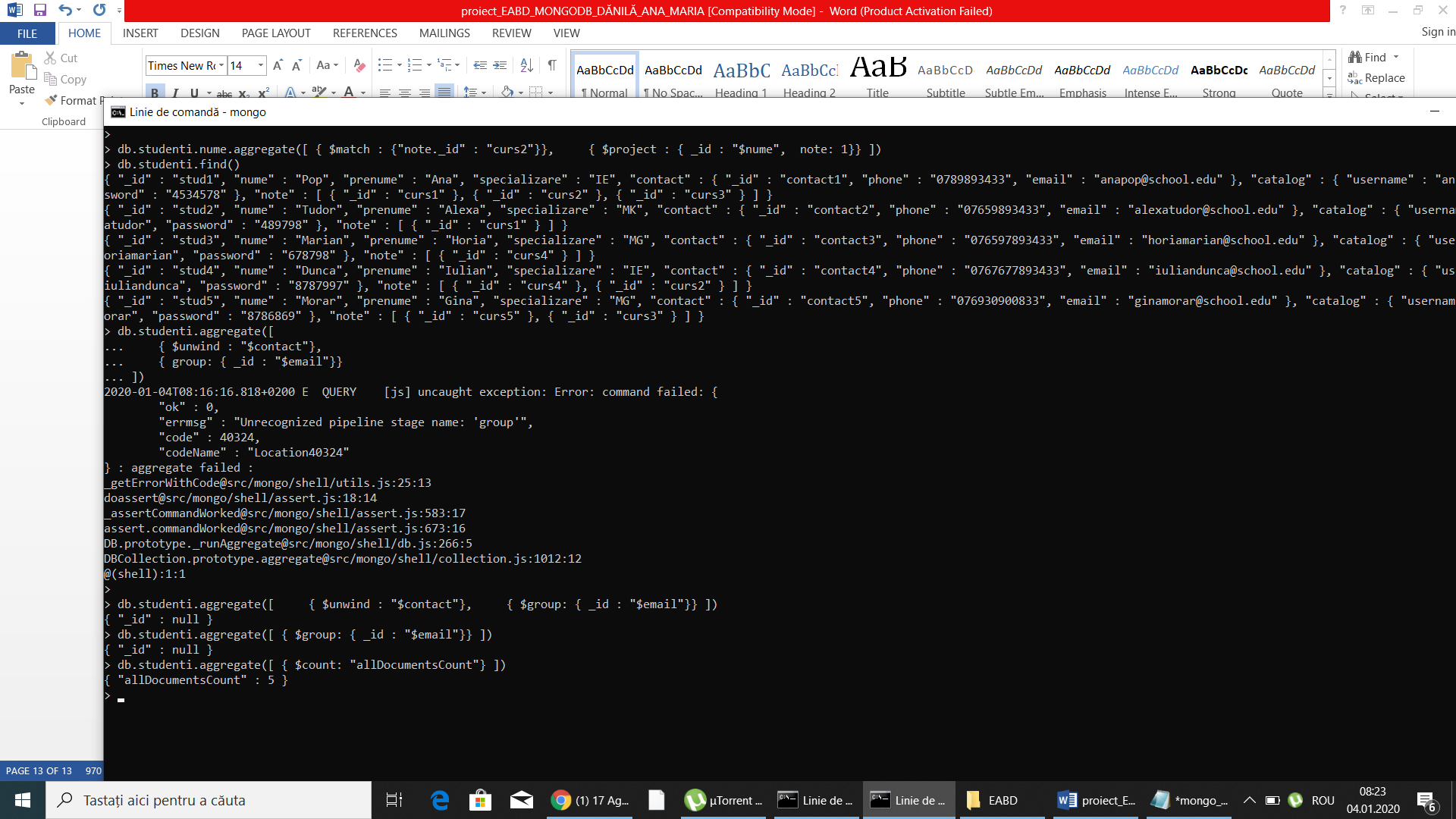
db.studenti.aggregate([{$group : {\_id : "$specializare", num\_studenti : {$sum : 1}}}])

7.**Să se afișeze toate cursurile care au taxe**

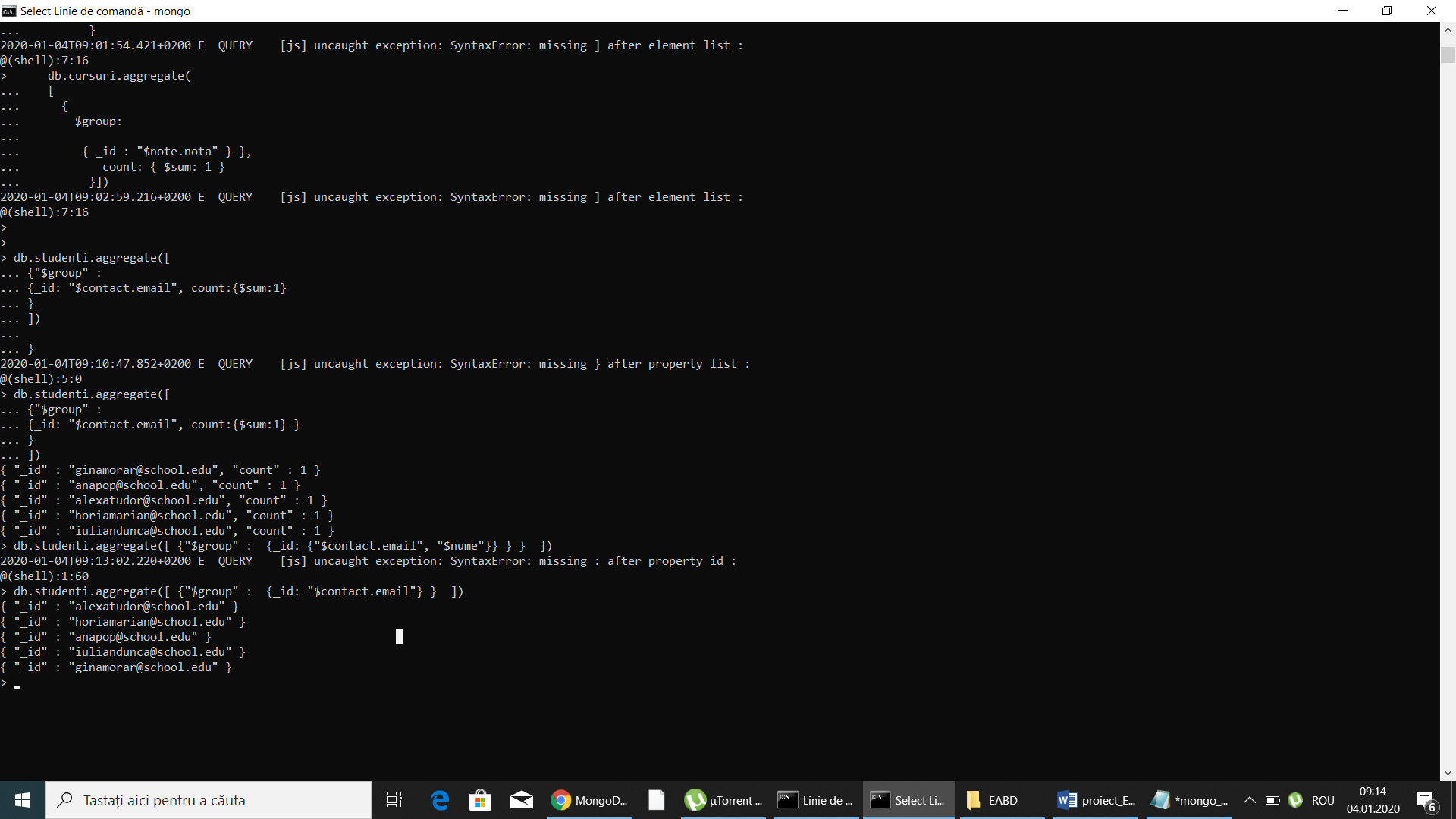
 db.cursuri.find( { taxe: { $exists: true } } )

**8.Să se afișeze cursurile la care s-a luat cel puțin o notă mai mare de 9.**

db.cursuri.find({ "note": { $elemMatch: { "nota": { $gt: 9 } } } }).pretty()

**9.Să se afișeze câți studenți avem în colecție**

db.studenti.aggregate([ { $count: "allDocumentsCount"} ])

**10.Să se afișeze email-urile studenților**

db.studenti.aggregate([ {"$group" : {\_id: "$contact.email"} } ])