Logo

Description automatically generated

ACADEMIA DE STUDII ECONOMICE BUCUREȘTI

FACULTATEA DE CIBERNETICĂ, STATISTICĂ ȘI INFORMATICĂ ECONOMICĂ

MASTER BAZE DE DATE – SUPORT PENTRU AACERI

**PROIECT**

**BAZE DE DATE NoSQL**

**Profesor coordonator:**

**Conf. Univ. Dr. Iuliana BOTHA (ȘIMONCA)**

**Student:**

**IORDACHE DENISA-MARIA**

**BUCUREȘTI**

**2022**

Cuprins

[Introducere 3](#_Toc119598914)

[1. Crearea bazei de date MongoBD și a colecției 4](#_Toc119598915)

[1.1. Crearea bazei de date 4](#_Toc119598916)

[1.2. Crearea colecției candidați 4](#_Toc119598917)

[2. Comenzi CRUD 5](#_Toc119598918)

[2.1. Inserare de documente în colecție 5](#_Toc119598919)

[2.2. Actualizarea documentelor din colecție (prin adăugarea de câmpuri și popularea lor) 9](#_Toc119598920)

[2.3. Regăsirea pe bază de diferite criterii a documentelor din colecție 10](#_Toc119598921)

[2.4. Ștergerea documentelor pe bază de criterii 14](#_Toc119598922)

[3. Exemplificarea agregărilor 15](#_Toc119598923)

# Introducere

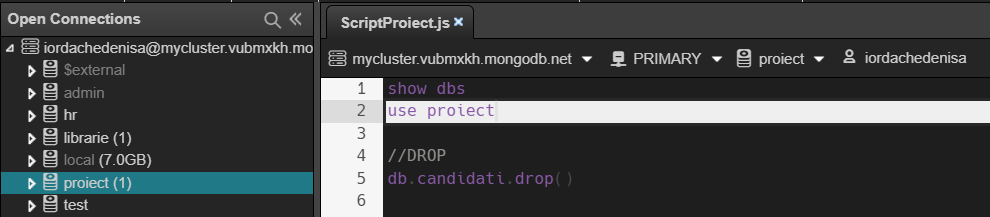
Baza mea de date conține o colecție numită „Candidați” cu date referitoare la candidații înscriși pentru obținerea unui loc la programele de licență din cadrul Academiei de Studii Economice din București.

Astfel, datele conținute de documente sunt următoarele: nume, prenume, telefon, email, notele de la bacalaureat, precum și media din liceu, acestea fiind relevante pentru calcularea mediei de admitere. Media de admitere se va calcula automat (80% contând media la bacalaureat și 20% media din liceu) la adăugarea unui document în colecție, lucru realizat printr-un update. De asemenea, dacă studentul încarcă eseul motivațional, câmpul corespunzător acestuia se va completa cu „Da”, în caz contrar cu nu. Sunt reținute și cele 3 opțiuni care fac parte din dosarul de admitere într-un câmp de tip array, acestea fiind relevante în ordinea scrierii lor. Ulterior, tot printr-un update, se va adăuga câmpul de documente necesare pentru înscriere, însă doar pentru candidații care au trimis deja eseul motivațional.

# Crearea bazei de date MongoBD și a colecției

## Crearea bazei de date

*use proiect*



## Crearea colecției candidați

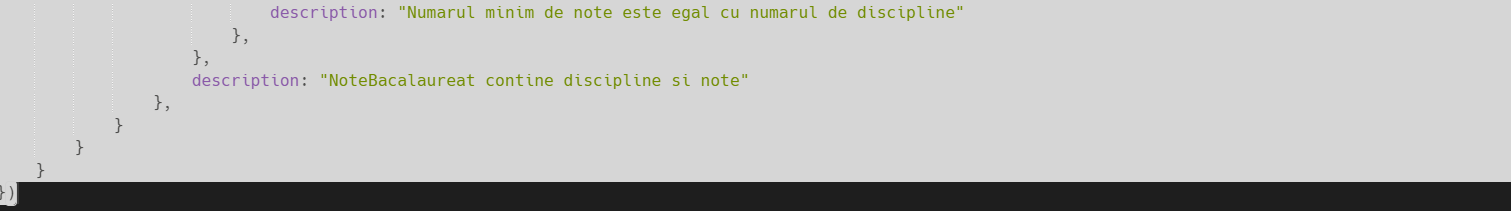
La crearea colecției există o serie de validări pe câmpuri, care pe lângă indicarea tipului de data, redă și niște reguli:

* numărul de telefon să aibă 10 cifre,
* emailul să fie de forma [ceva@gmail.com](mailto:ceva@gmail.com) sau [ceva@yahoo.com](mailto:ceva@yahoo.com),
* eseul motivațional va primi doar valorile da și nu,
* opțiunile universitare trebuie să fie în număr de 3,
* notele și disciplinele de la bacalaureat trebuie să fie tot în număr de 3, iar notele de la 5 la 10,
* media din liceu trebuie să aibă valori între 5 și 10

De asemenea, câmpurile marcate în required sunt obligatorii la crearea unui document.

*O imagine care conține text

Descriere generată automat* *O imagine care conține text

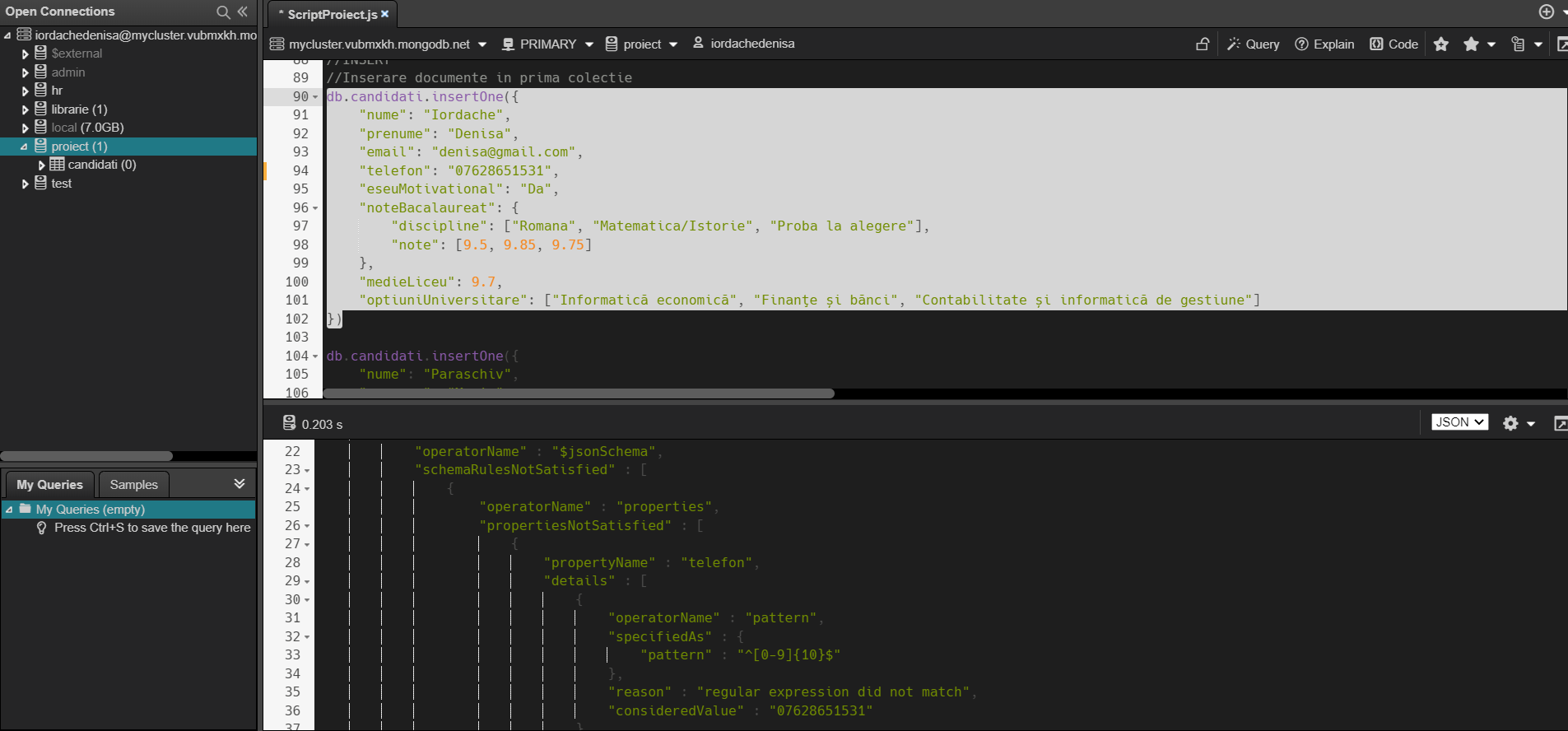
Descriere generată automat* 

# Comenzi CRUD

## Inserare de documente în colecție

Am folosit atât metoda insertOne (inserarea unui singur document), cât și insertMany (inserarea mai multor documente). La fiecare document am specificat cel puțin câmpurile obligatorii.

* **Pentru a testa funcționalitatea validărilor, o să inserez un număr de telefon format din 11 cifre pentru a afișa eroarea.**



* **Inserarea unui document în colecție**

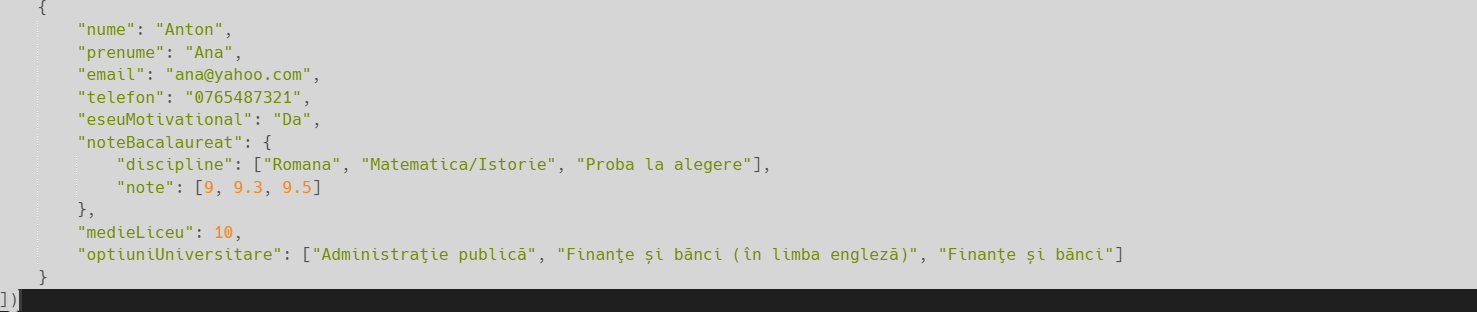


* **Inserarea mai multor documente în colecție**

O imagine care conține text

Descriere generată automat O imagine care conține text

Descriere generată automat O imagine care conține text

Descriere generată automat 

Rezultat: 

* **Adăugarea unui document din aplicația creată (unde se calculează automat câmpurile medieBac și medieAdmitere**



## Actualizarea documentelor din colecție (prin adăugarea de câmpuri și popularea lor)

* **Să se adauge câmpul medieBac și să se calculeze valoarea lui ca medie aritmetică a notelor din bac, rotunjită la 2 zecimale.**

*db.candidati.updateMany({}, [{ $set: { "medieBac": { $round: [{ $avg: "$noteBacalaureat.note" }, 2] } } }])*

* **Să se adauge câmpul medieAdmitere și să se calculeze valoarea lui ca 0.8\*medieBac + 0.2\*medieLiceu, rotunjită la 2 zecimale.**

*db.candidati.updateMany({}, [{ $set: { "medieAdmitere": { $round: [{ $add: [{ $multiply: ["$medieBac", 0.8] }, { $multiply: ["$medieLiceu", 0.2] }] }, 2] } } }])*

* **Să se scrie comanda pentru ștergerea unuia din câmpurile adăugate anterior.**

*db.candidati.updateMany({}, { $unset: { "medieBac": null } })*

* **Să se adauge câmpul documenteIncarcate care este de tip colecție ce conține câmpurile CopieBuletin, CopieCertificatNastere, DiplomaBacalaureat, FoateMatricola și AdeverintaMedicala, toate populate cu „Nu” inițial, doar pentru candidații care au încărcat deja eseul motivațional.**

*db.candidati.updateMany({ "eseuMotivational": "Da" }, [{ $set: { "documenteIncarcate": Object({ "CopieBuletin": "Nu", "CopieCertificaNastere": "Nu", "DiplomaBacalaureat": "Nu", "FoaieMatricola": "Nu", "AdeverintaMedicala": "Nu" }) } }])*

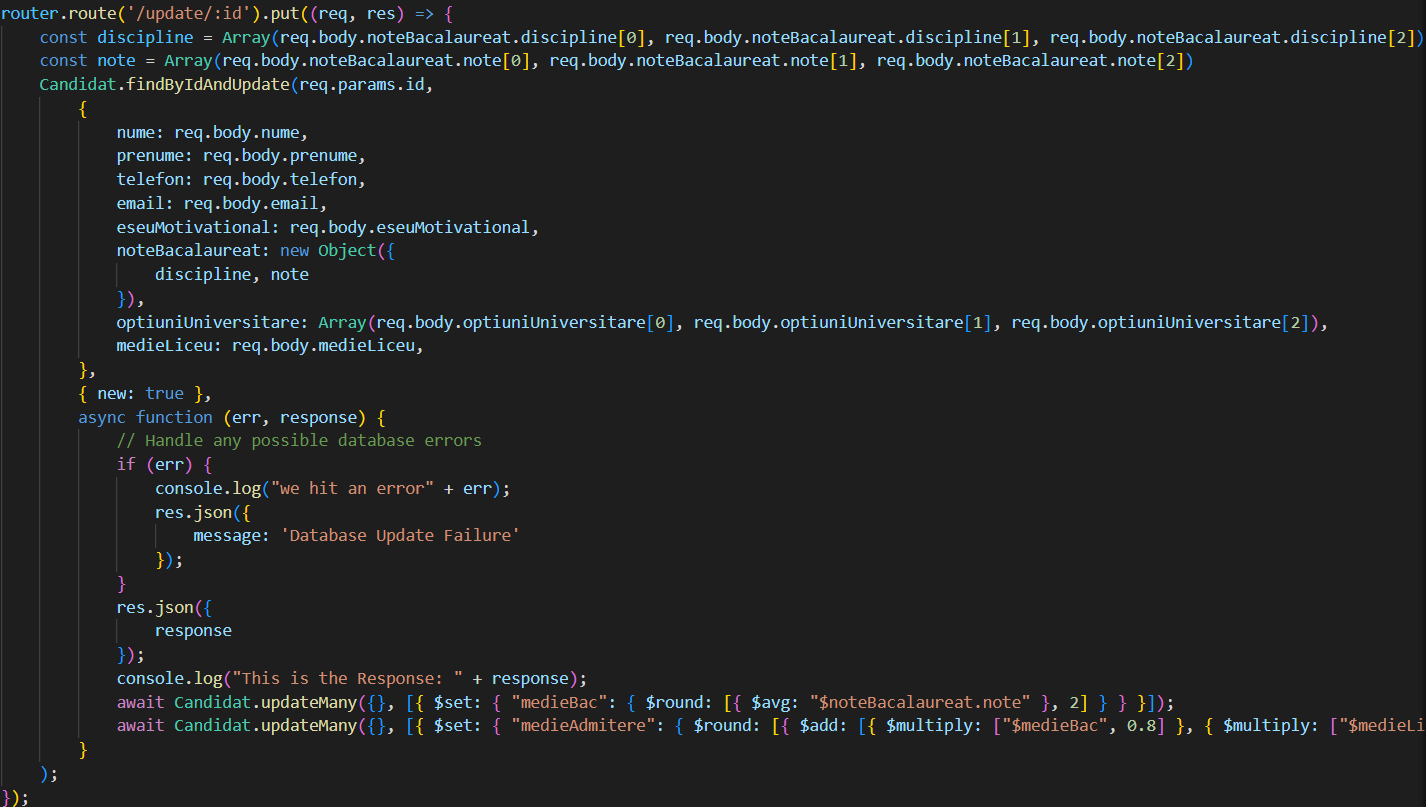
* **Pentru candidatul al cărui număr de telefon se termină în \*\*\*\*\*\*\*153 să se noteze corespunzător atașarea copiei de buletin, copiei certificatului de naștere si adeverinței medicale.**

*db.candidati.updateMany(*

*{ telefon: { $regex: /153$/ } },*

*{ $set: { "documenteIncarcate.CopieBuletin": "Da", "documenteIncarcate.CopieCertificaNastere": "Da", "documenteIncarcate.AdeverintaMedicala": "Da" } })*

* **Actualizarea după id în aplicație (se actualizează și câmpurile medieBac și medieAdmitere automat)**



## Regăsirea pe bază de diferite criterii a documentelor din colecție

* **Regăsirea tuturor candidaților, fără afișarea id-ului.**

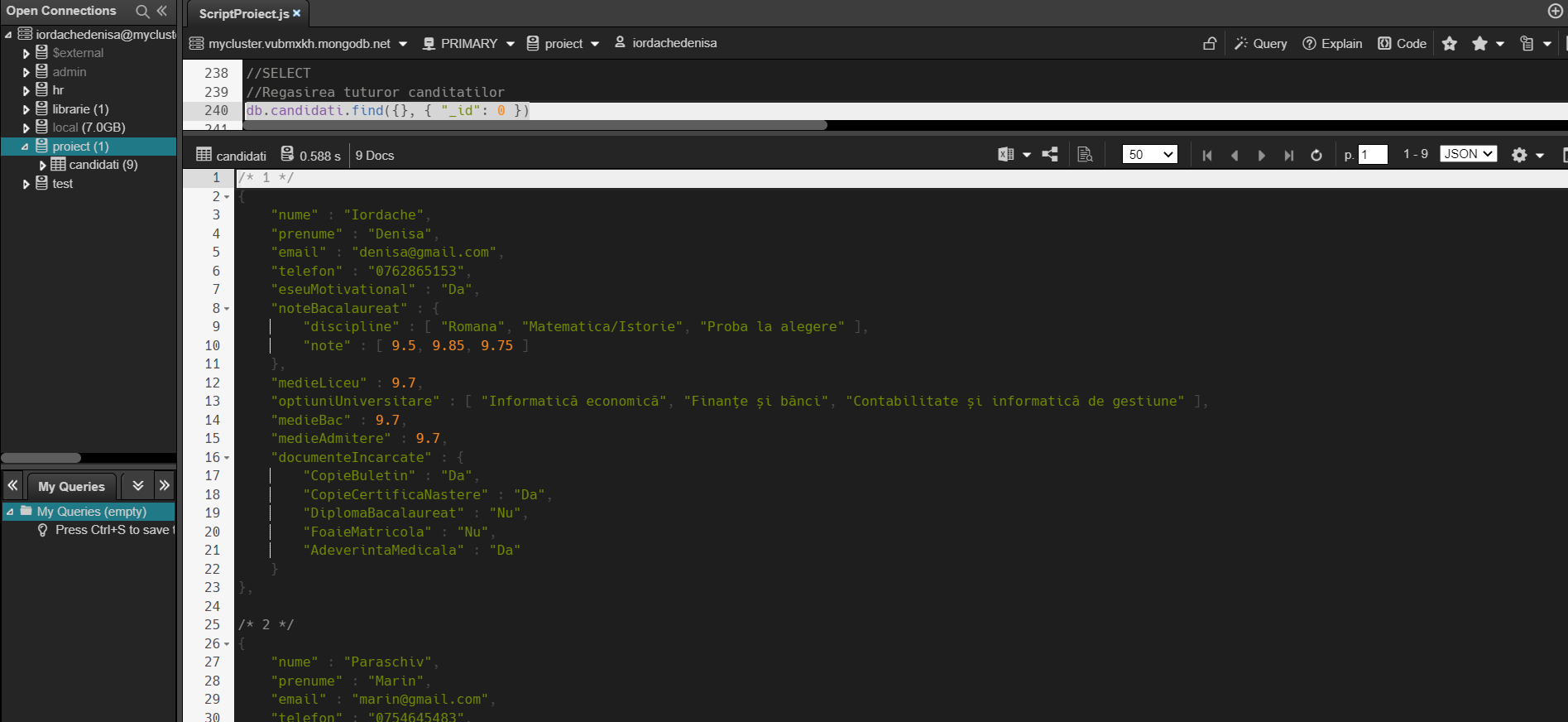
**În aplicație:**

O imagine care conține text

Descriere generată automat

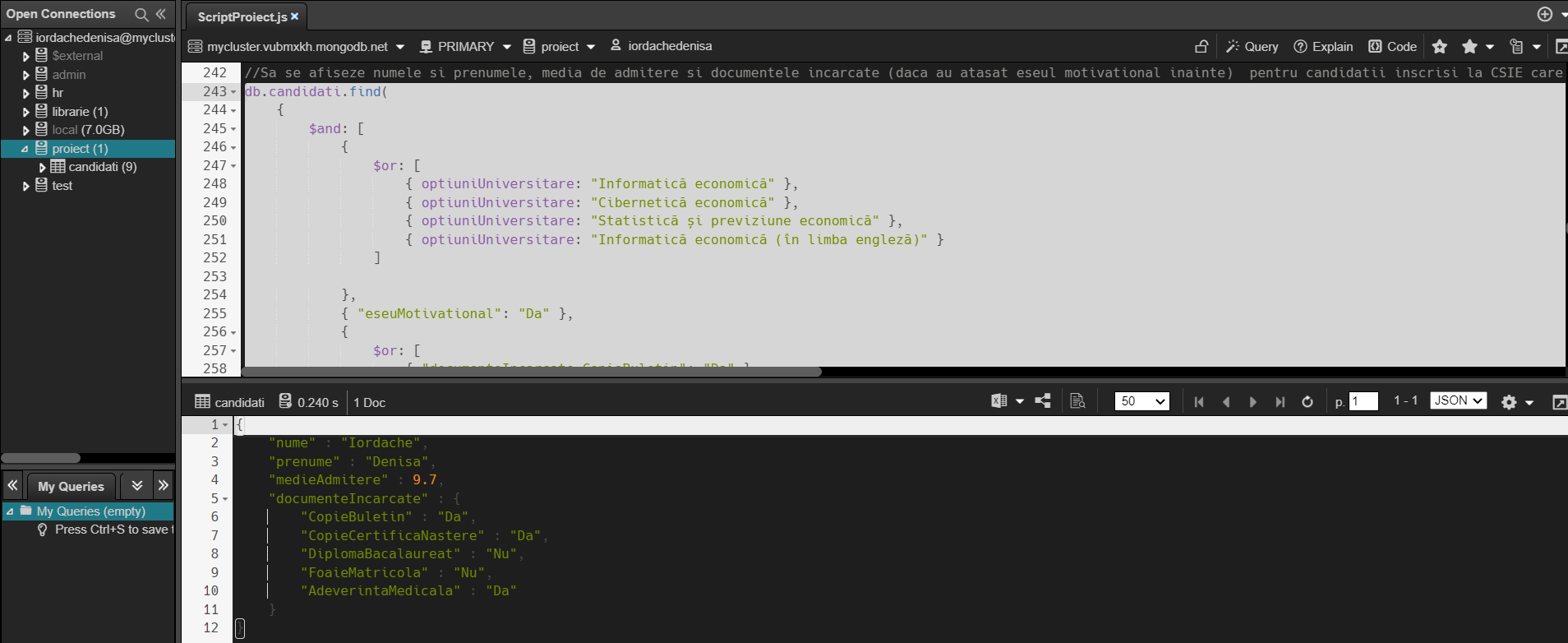
**În NoSQLBooster:**

*db.candidati.find({}, { "\_id": 0 })*

**

* **Să se afișeze numele, prenumele, media de admitere și documentele încărcate (dacă au atașat eseul motivațional înainte) pentru candidații înscriși la CSIE care au cel puțin un document încărcat.**

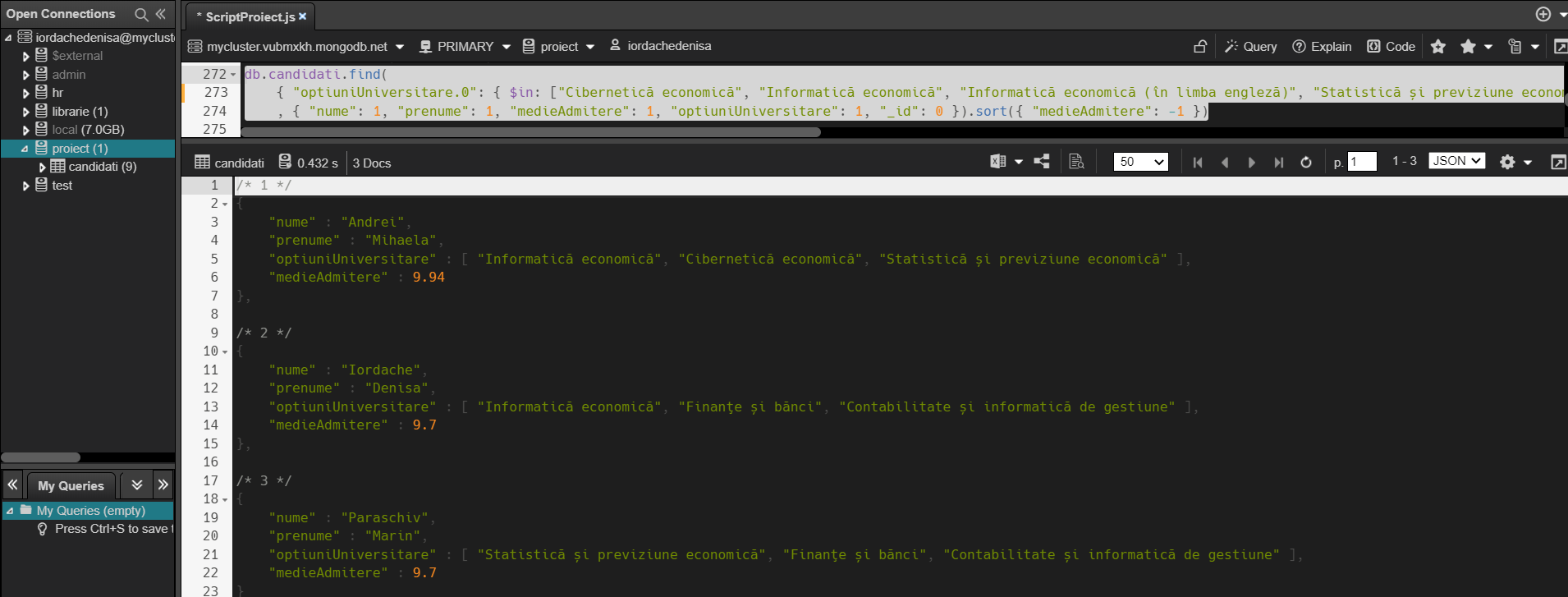




* **Afișarea numelui, prenumelui, mediei de admitere și a opțiunilor universitare pentru cei care au ca și primă opțiune o specializare din CSIE in ordine descrescătoare mediei de admitere.**

*db.candidati.find({ "optiuniUniversitare.0": { $in: ["Cibernetică economică", "Informatică economică", "Informatică economică (în limba engleză)", "Statistică și previziune economică"] } }*

*, { "nume": 1, "prenume": 1, "medieAdmitere": 1, "optiuniUniversitare": 1, "\_id": 0 }).sort({ "medieAdmitere": -1 })*

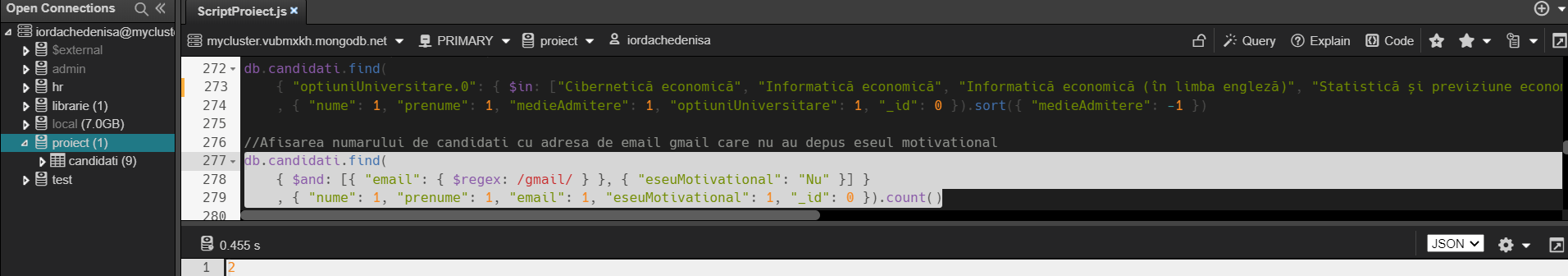


* **Afișarea numărului de candidați cu adresa de gmail care nu au depus eseul motivațional.**

*db.candidati.find(*

*{ $and: [{ "email": { $regex: /gmail/ } }, { "eseuMotivational": "Nu" }] }*

*, { "nume": 1, "prenume": 1, "email": 1, "eseuMotivational": 1, "\_id": 0 }).count()*



* **Afișarea candidaților înscriși la programe cu predate in limba străină in ordine descrescătoare notei din bac la romană.**

*db.candidati.find(*

*{*

*$or: [*

*{ optiuniUniversitare: { $regex: /(în limba engleză)/ } },*

*{ optiuniUniversitare: { $regex: /(engleză - franceză)/ } },*

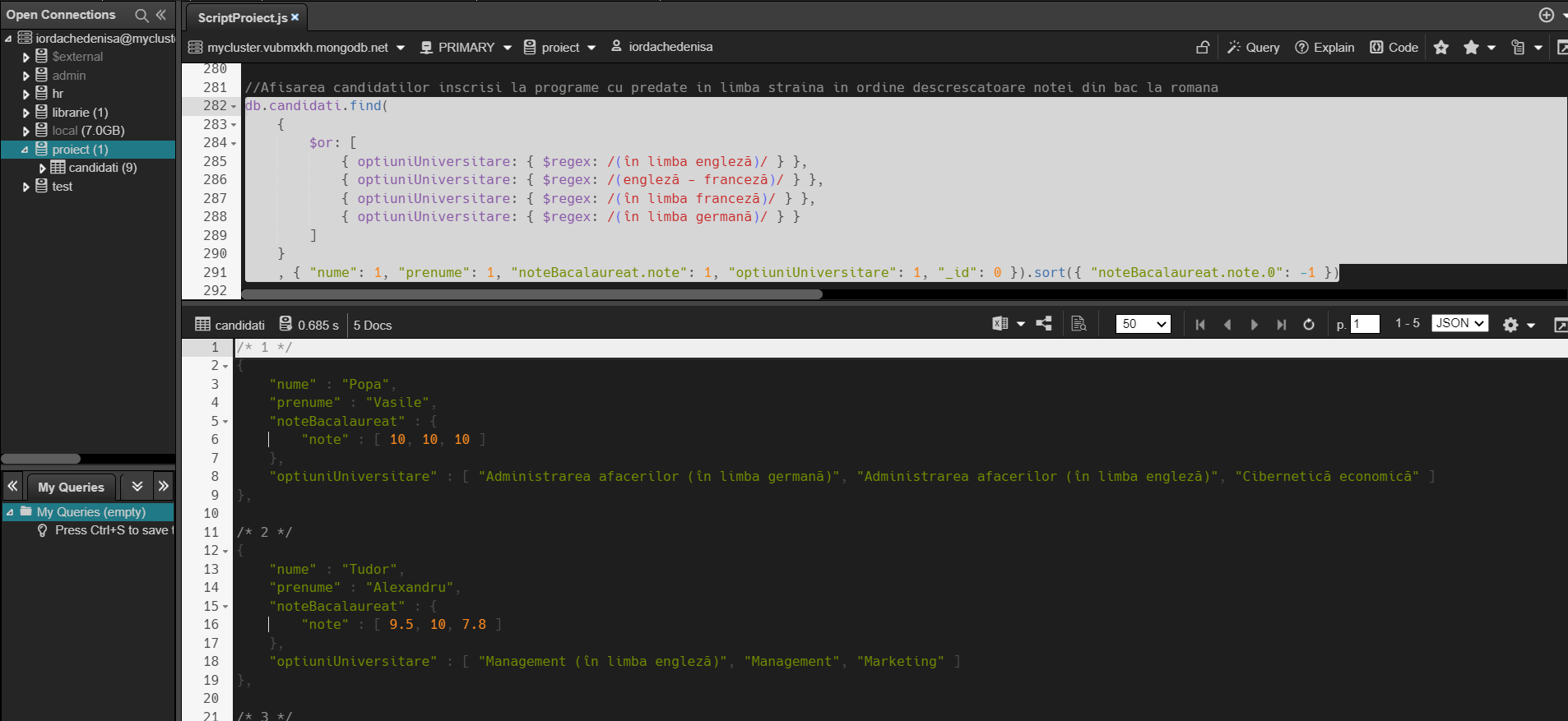
*{ optiuniUniversitare: { $regex: /(în limba franceză)/ } },*

*{ optiuniUniversitare: { $regex: /(în limba germană)/ } }*

*]*

*}*

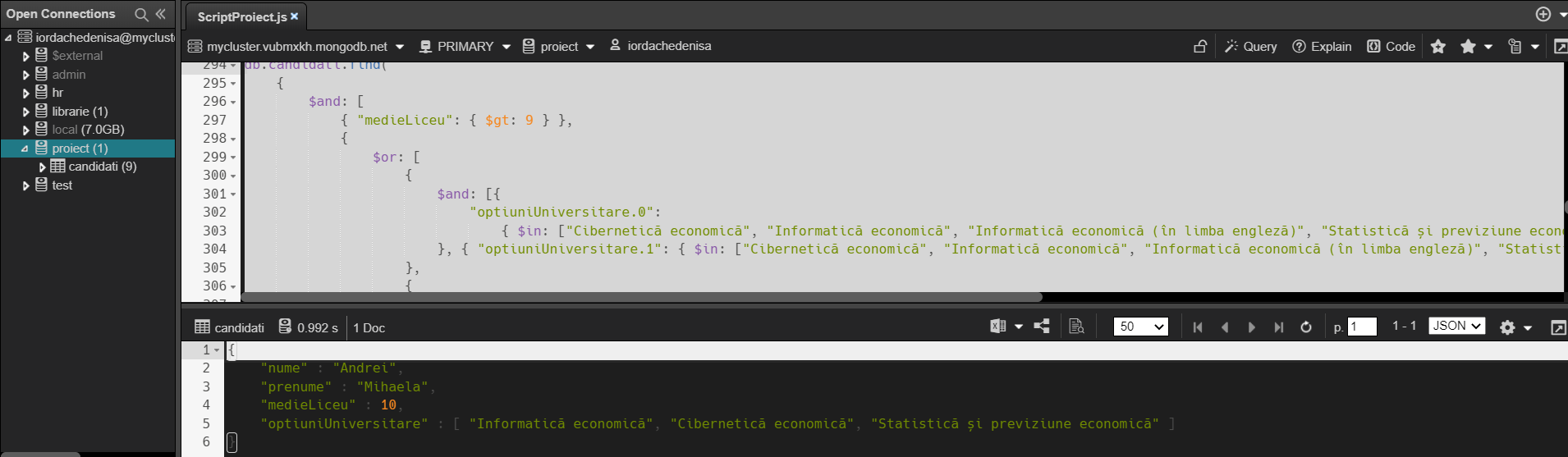
*, { "nume": 1, "prenume": 1, "noteBacalaureat.note": 1, "optiuniUniversitare": 1, "\_id": 0 }).sort({ "noteBacalaureat.note.0": -1 })*



* **Afișarea candidaților care au avut media din liceu peste 9 și au aplicat la minim 2 specializări din CSIE.**

O imagine care conține text

Descriere generată automat

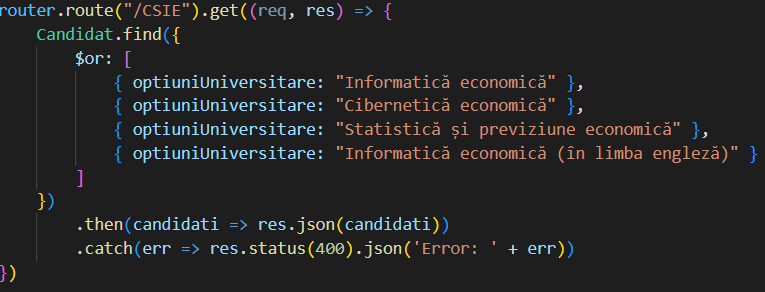
**

* **Regăsirea unui candidat după id în aplicație**

O imagine care conține text

Descriere generată automat

* **Gruparea specializărilor în facultăți (am procedat astfel pentru toate facultățile pentru o gestionare mai ușoară în frontend)**



## Ștergerea documentelor pe bază de criterii

* **Ștergerea candidaților care nu au depus eseul motivațional.**

*db.candidati.deleteMany({ "eseuMotivational": "Nu" })*

* **Eliminarea candidaților înscriși la FAMP care au media de admitere mai mică de 8.5.**

*db.candidati.deleteMany({*

*$and: [{*

*$or: [*

*{ optiuniUniversitare: "Administrație publică" },*

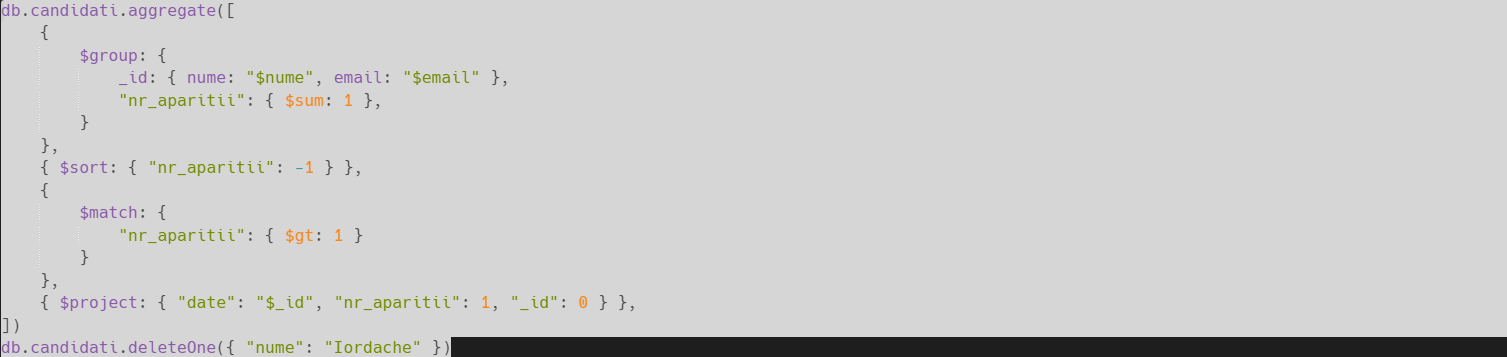
*{ optiuniUniversitare: "Resurse umane" },*

*]*

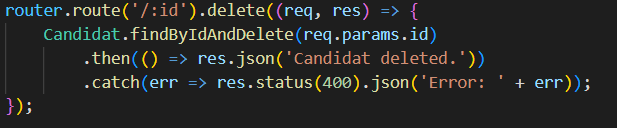
*}, { "medieAdmitere": { $lt: 8.5 } }]*

*})*

* **Eliminarea duplicatelor din colecție (prin agregare le găsesc, iar folosind metoda deleteOne le șterg).**



* **Ștergerea unui document după id în aplicație.**



# Exemplificarea agregărilor

* **Să se grupeze candidații după prima opțiune aleasă și să se calculeze numărul de persoane care au ales-o, care este media de admitere cea mai mare, cea mai mică si media pentru fiecare grup, afișându-se descrescător după media de admitere medie.**

*db.candidati.aggregate([*

*{*

*$group: {*

*\_id: { $arrayElemAt: ["$optiuniUniversitare", 0] },*

*"maxMedieAdmitere": { $max: "$medieAdmitere" },*

*"avgMedieAdmitere": { $avg: "$medieAdmitere" },*

*"minMedieAdmitere": { $min: "$medieAdmitere" },*

*"candidati": { $sum: 1 },*

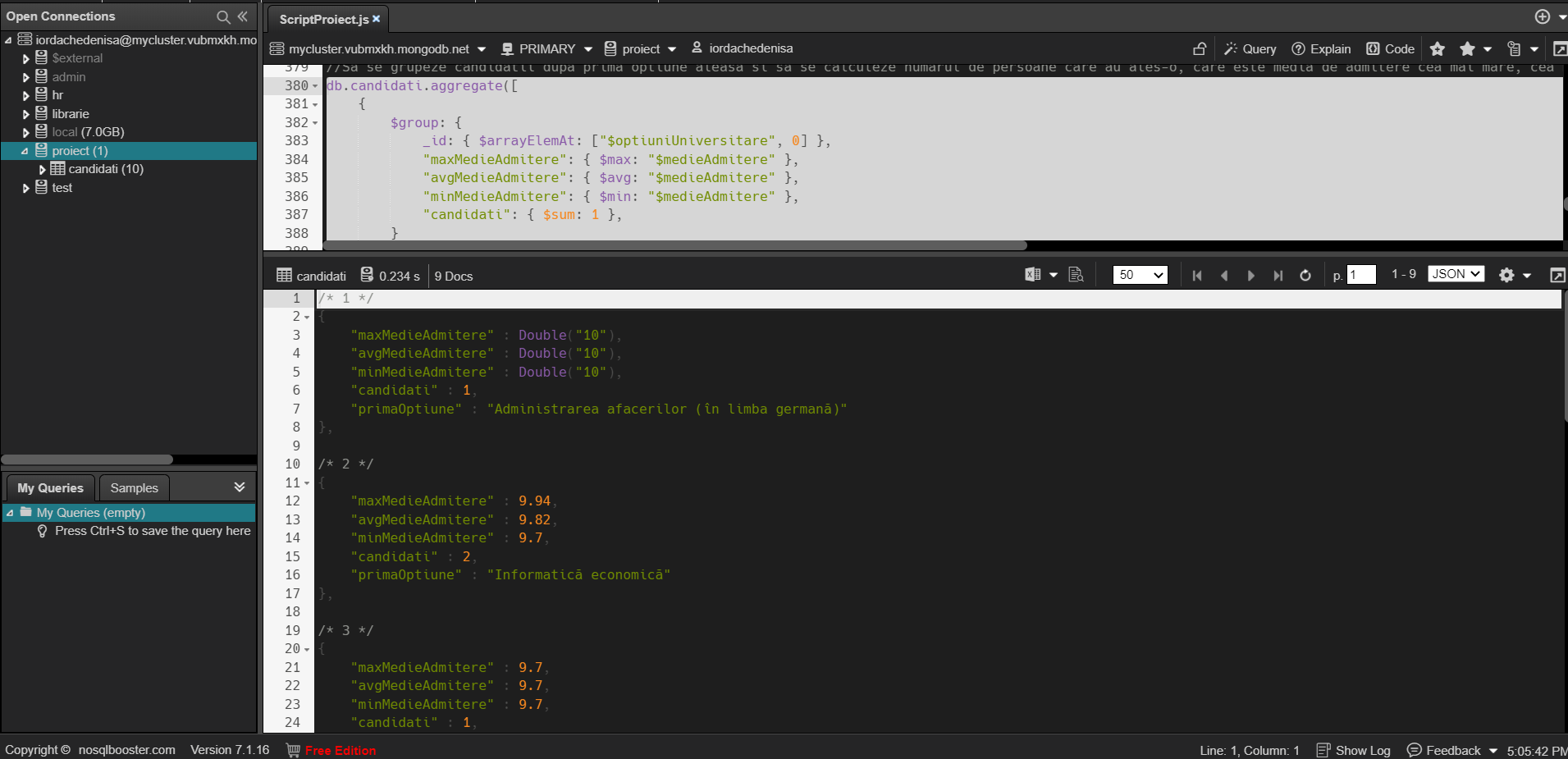
*}*

*},*

*{ $sort: { "avgMedieAdmitere": -1 } },*

*{ $project: { "primaOptiune": "$\_id", "maxMedieAdmitere": 1, "avgMedieAdmitere": 1, "minMedieAdmitere": 1, "candidati": 1, "\_id": 0 } }*

*]);*



* **Să se calculeze numărul candidaților la FABBV care nu au depus încă niciun document și să se grupeze pe fiecare specializare în parte.**

*db.candidati.aggregate([*

*{*

*$match: {*

*$and: [{ $or: [{ "optiuniUniversitare": "Finanțe și bănci" }, { "optiuniUniversitare": "Finanțe și bănci (în limba engleză)" }] },*

*{ $and: [{ "documenteIncarcate.CopieBuletin": "Nu" }, { "documenteIncarcate.CopieCertificaNastere": "Nu" }, { "documenteIncarcate.DiplomaBacalaureat": "Nu" }, { "documenteIncarcate.FoaieMatricola": "Nu" }, { "documenteIncarcate.AdeverintaMedicala": "Nu" }] }]*

*}*

*},*

*{ $project: { "optiuniUniversitare": 1 } },*

*{ $unwind: "$optiuniUniversitare" },*

*{*

*$group: {*

*\_id: "$optiuniUniversitare",*

*"candidati": { $sum: 1 },*

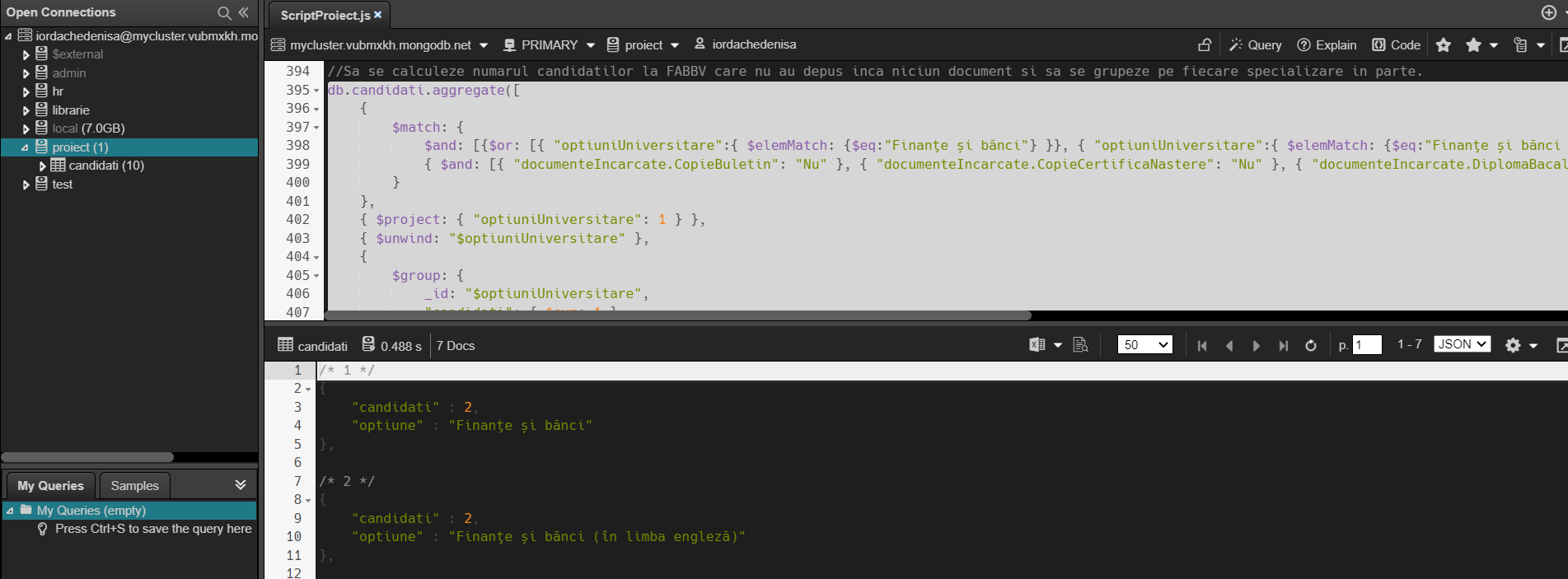
*}*

*},*

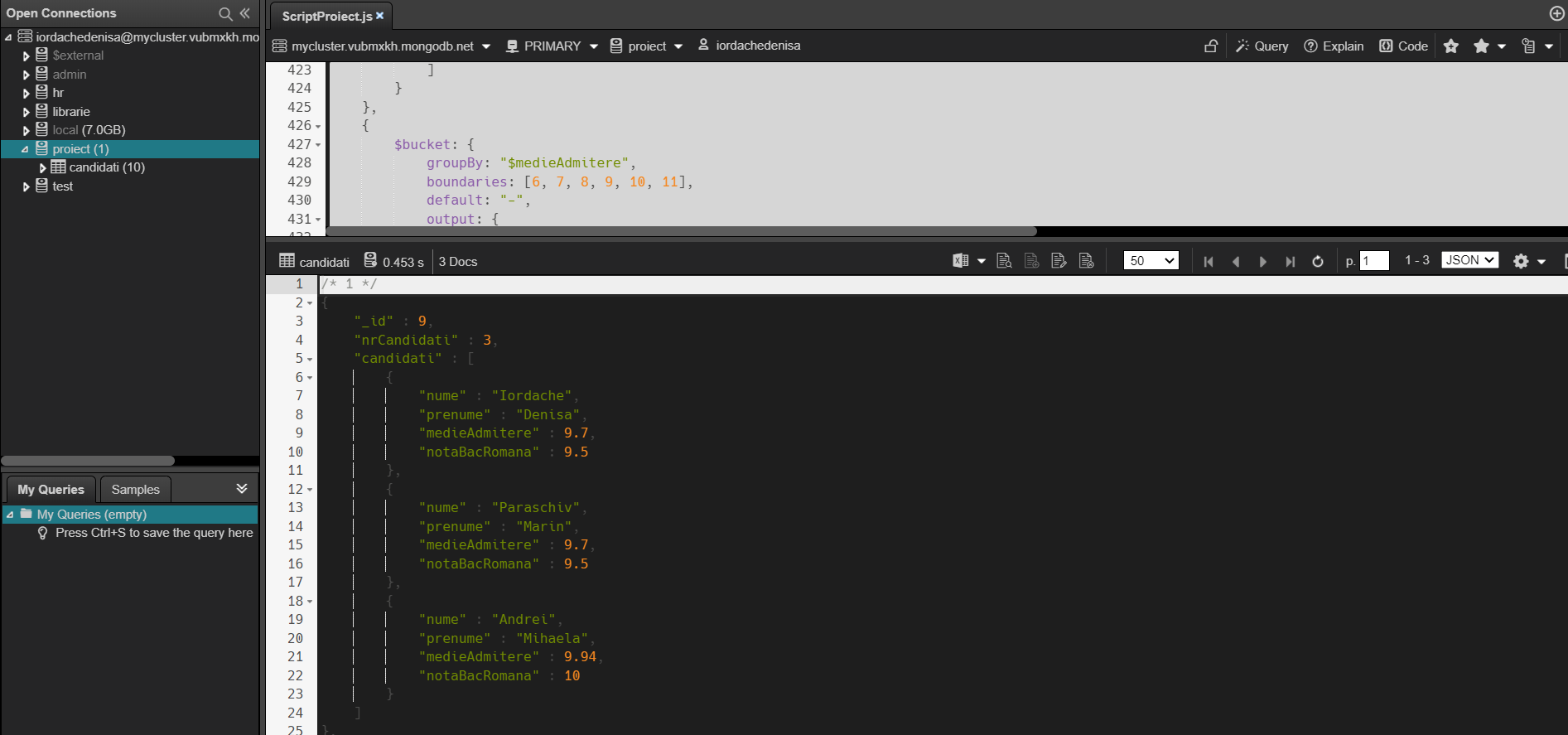
*{ $sort: { "candidati": -1 } },*

*{ $project: { "optiune": "$\_id", "candidati": 1, "\_id": 0 } }*

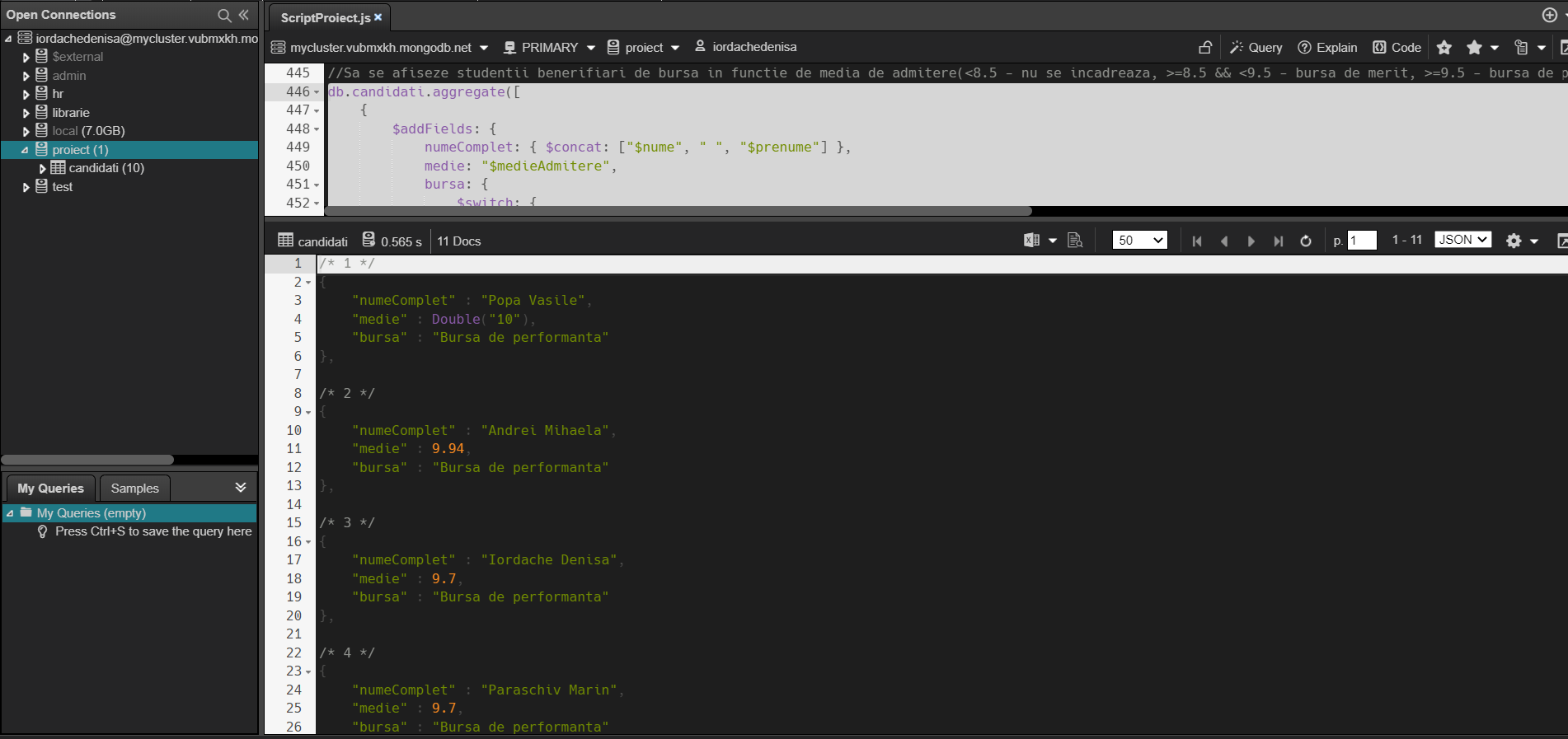
*]);*



* **Afișarea candidaților la CSIE folosind intervale de valori (bucket). Se va afișa numele, prenumele, media de admitere și media din bac la romana (pentru departajare).**

**

* **Să se afișeze studenții beneficiari de bursă in funcție de media de admitere(<8.5 - nu se încadrează, >=8.5 && <9.5 - bursă de merit, >=9.5 - bursă de performantă)**

**

* **Afișarea candidaților la CSIE care au aceeași medie de admitere în ordine descrescătoare a notei la bacalaureat (pentru departajare).**

*db.candidati.aggregate([*

*{*

*$match: {*

*"optiuniUniversitare.0": { $in: ["Cibernetică economică", "Informatică economică", "Informatică economică (în limba engleză)", "Statistică și previziune economică"] }*

*}*

*},*

*{ $sort: { "medieBac": -1 } },*

*{*

*$group: {*

*\_id: "$medieAdmitere",*

*"nr\_aparitii": { $sum: 1 },*

*"candidati": { $push: { $concat: ["$nume", " ", "$prenume"] } }*

*}*

*},*

*{ $sort: { "nr\_aparitii": -1 } },*

*{ $project: { "medieAdmitere": "$\_id", "candidati": 1, "nr\_aparitii": 1, "\_id": 0 } },*

*{ $limit: 1 }*

*]);*

