Serkan Berk Bilgiç

71571

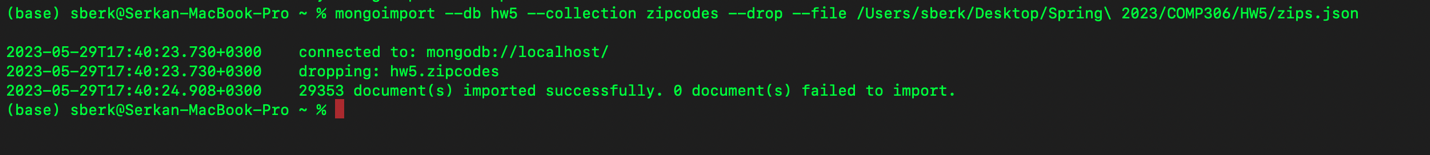
HW5 Report

**Question 1-**

* use hw5
* mongoimport --db hw5 --collection zipcodes --drop --file /Users/sberk/Desktop/Spring\ 2023/COMP306/HW5/zips.json
* show collections
* db.zipcodes.find().limit(15)

**A screen shot of a computer

Description automatically generated with medium confidence**

A screen shot of a computer

Description automatically generated with medium confidence

A screenshot of a computer code

Description automatically generated with medium confidence

A screenshot of a computer code

Description automatically generated with low confidence

**Question 2-**

* db.zipcodes.find({ "pop": { $gt: 50000 }, "state": "CA", "loc.1": {$gt: 35} } ).sort({"pop": -1}).limit(5)

A screenshot of a computer program

Description automatically generated with medium confidence

**Question 3-**

* db.zipcodes.find({$or: [{state: {$ne: "CA"} }, {$and: [{"loc.0": {$lt: -120}}, {"loc.1": {$gt: 40}}]}]})
* db.zipcodes.find({ $or: [ { state: { $ne: "CA" } }, { "loc.0": { $lt: -120 }, "loc.1": { $gt: 40 } }] })

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with low confidence

A screenshot of a computer program

Description automatically generated with low confidence

**Question 4-**

* db.zipcodes.aggregate([{$group: {\_id: "$city", totalPopulation: {$sum: "$pop"} } }, {$sort: {totalPopulation: -1}}, {$limit: 5}])

A screen shot of a computer

Description automatically generated with medium confidence

**Question 5-**

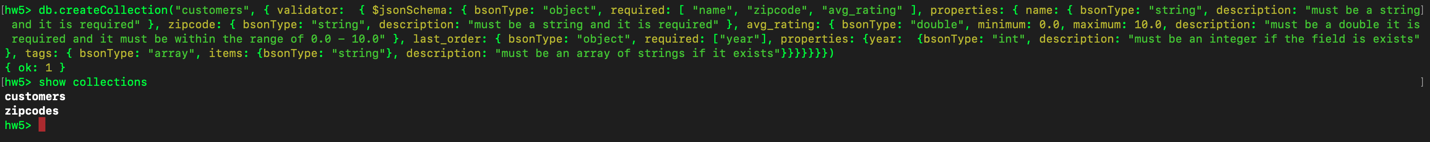
* db.zipcodes.aggregate([{$group: {\_id: "$state", count: {$sum: 1} } }, {$match: {count: {$gt: 300, $lt: 500 }}}, { $project: { state: "$\_id", \_id: 0, count: 1 } }, {$sort: {count: 1} }])

A picture containing screenshot

Description automatically generated

**Question 6-**

* db.createCollection("customers", {
* ... validator: {
* ... $jsonSchema: {
* ... bsonType: "object",
* ... required: [ "name", "zipcode", "avg\_rating" ],
* ... properties: {
* ... name: {
* ... bsonType: "string",
* ... description: "must be a string and it is required"
* ... },
* ... zipcode: {
* ... bsonType: "string",
* ... description: "must be a string and it is required"
* ... },
* ... avg\_rating: {
* ... bsonType: "double",
* ... minimum: 0.0,
* ... maximum: 10.0,
* ... description: "must be a double it is required and it must be within the range of 0.0 - 10.0"
* ... },
* ... last\_order: {
* ... bsonType: "object",
* ... required: ["year"],
* ... properties: {
* ... year: {
* ... bsonType: "int",
* ... description: "must be an integer if the field is exists"
* ... },
* ... tags: {
* ... bsonType: “array”,
* … items: {
* … bsonType: “string”
* … },
* ... description: "must be an array of strings if it exists"
* ... }
* ... }
* ... }
* ... }
* ... }
* ... }
* ... }
* ... )



**Question 7-**

* db.customers.insertMany([
* { "name": "Serkan Berk Bilgiç", "zipcode": "99503", "avg\_rating": 8.3 },
* { "name":"Andre T.", "zipcode":"90025", "avg\_rating": 3.5, "last\_order":{ "year": 2009 } },
* { "name": "Bela T.", "zipcode": "33126", "avg\_rating": 4.9, "last\_order": { "year": 2019, "tags": ["art", "melancholy"] } },
* { "name": "Nuri Bilge C.", "zipcode": "90010", "avg\_rating": 6.5, "last\_order": { "year": 3005 } }
* ] )

A screen shot of a computer

Description automatically generated with medium confidence

**Question 8-**

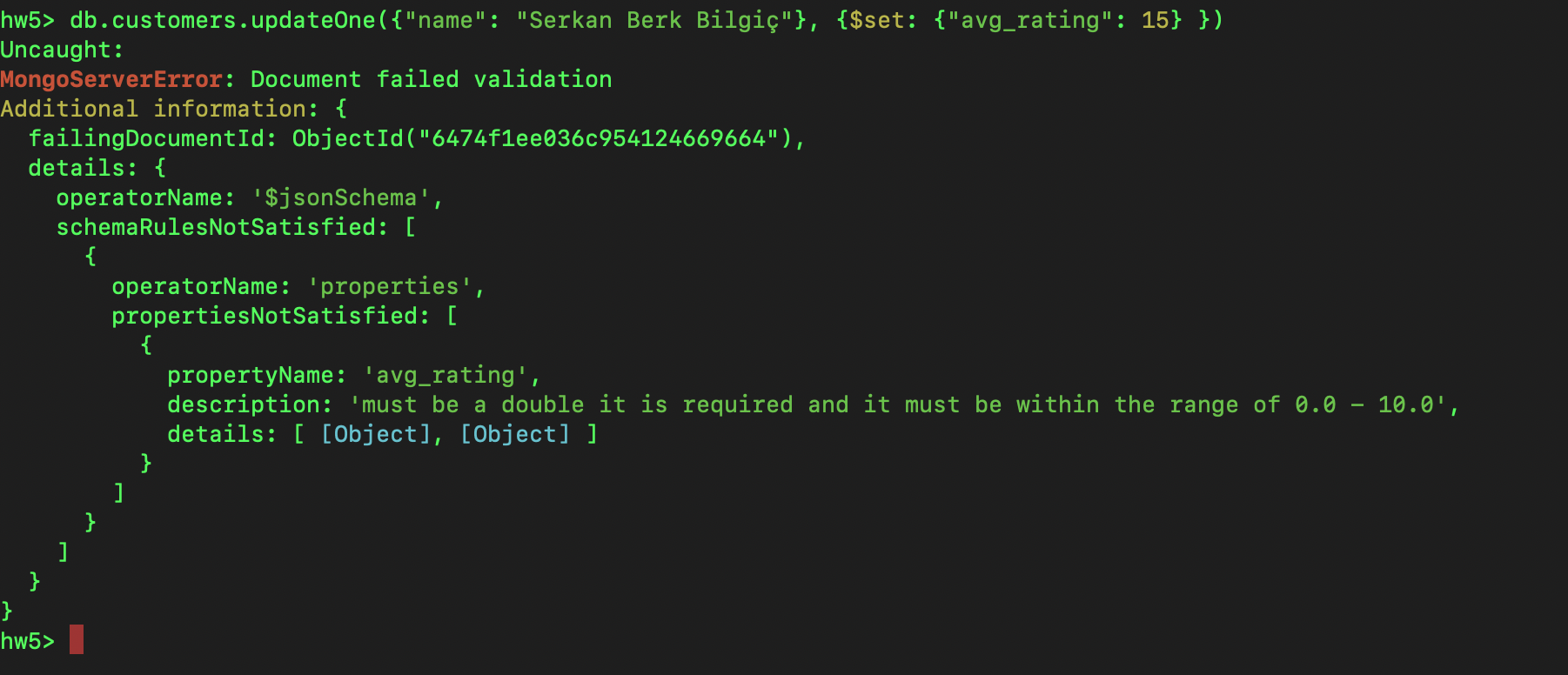
* db.customers.deleteMany({"last\_order.year": {$gt: 2023}})

A picture containing text, screenshot

Description automatically generated

**Question 9-**

* db.customers.updateOne({"name": "Serkan Berk Bilgiç"}, {$set: {"avg\_rating": 15} })
* As we can see from the error message it does not update avg\_rating because we had an requirement while we design the schema validation that avg\_rating must be between 0.0 and 10.0
* It is a good solution to ensure consistency of our data and helps to prevent wrong data entries



**Question 10-**

* db.customers.aggregate({$lookup: {from: "zipcodes", localField: "zipcode", foreignField: "\_id", as: "customer\_zipcodes"}}, {$project: {\_id: 0, name: 1, city: "$customer\_zipcodes.city", state: "$customer\_zipcodes.state", zipcode: "$customer\_zipcodes.\_id"}}, {$sort: {name: 1}})

A screenshot of a computer

Description automatically generated with medium confidence

**Question 11-**

* mongoexport --db hw5 --collection customers --out customers.json

