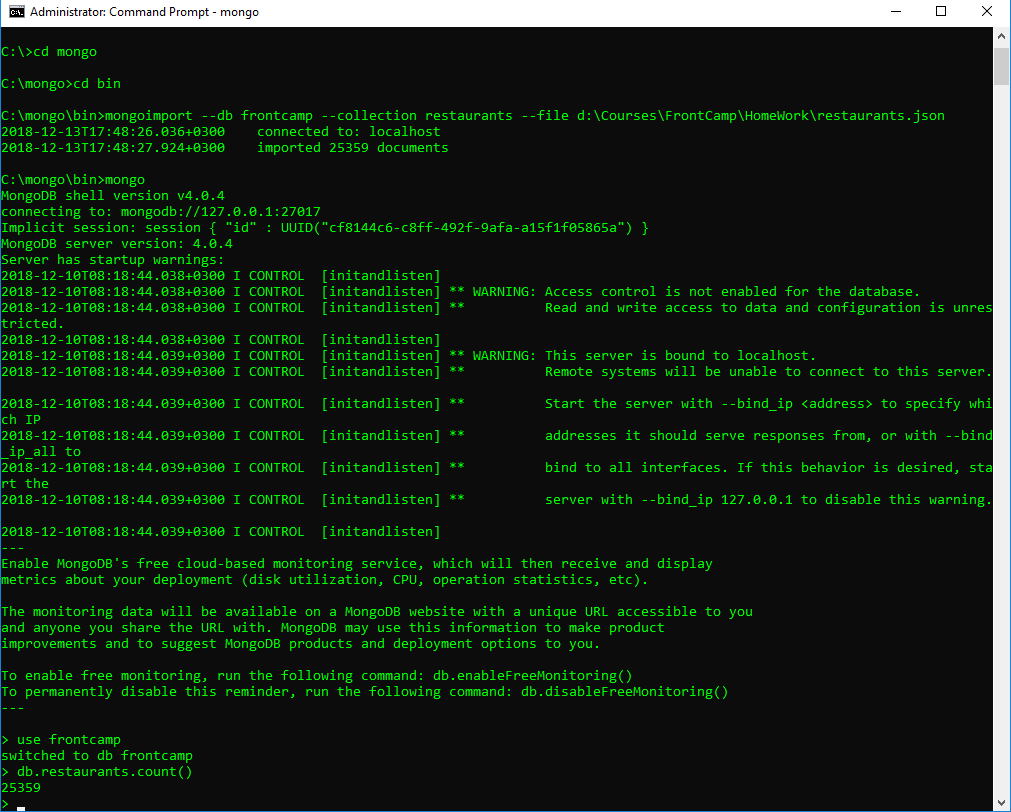
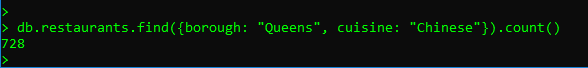
MongoDB part1.

Start MongoDB and import restaurants.json



1. How many “Chinese” (cuisine) restaurants are in “Queens” (borough)?

**  
  
Query:**

>db.restaurants.find({borough: "Queens", cuisine: "Chinese"}).count()

**Result:**

728

2. What is the \_id of the restaurant which has the grade with the highest ever score?



**Query:**

> db.restaurants.find({}, {\_id: 1}).sort({"grades.score": -1}).limit(1)

**Result:**

{ "\_id" : ObjectId("5c12713a9afd28ad5cf5be8e") }

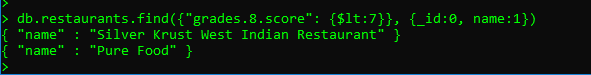
3.Add a grade { grade: "A", score: 7, date: ISODate() } to every restaurant in “Manhattan” (borough).



**Query:**

>db.restaurants.updateMany({ borough: "Manhattan" }, {$push: {"grades" : { grade:"A", score:7, date: ISODate()}}})

4. What are the names of the restaurants which have a grade at index 8 with score less then 7? Use projection to include only names without \_id.



**Query:**

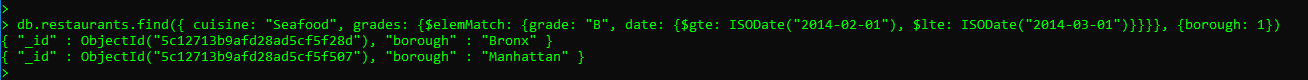
>db.restaurants.find({"grades.8.score": {$lt:7}}, {\_id:0, name:1})

**Response:**

{ "name" : "Silver Krust West Indian Restaurant" }

{ "name" : "Pure Food" }

5. What are \_id and borough of “Seafood” (cuisine) restaurants which received at least one “B” grade in period from 2014-02-01 to 2014-03-01? Use projection to include only \_id and borough.



**Query:**  
>db.restaurants.find({ cuisine: "Seafood", grades: {$elemMatch: {grade: "B", date: {$gte: ISODate("2014-02-01"), $lte: ISODate("2014-03-01")}}}}, {borough: 1})

**Response:**

{ "\_id" : ObjectId("5c12713b9afd28ad5cf5f28d"), "borough" : "Bronx" }

{ "\_id" : ObjectId("5c12713b9afd28ad5cf5f507"), "borough" : "Manhattan" }

**Task4 Indexes.**

1. Create an index which will be used by this query and provide proof (from explain() or Compass UI) that the index is indeed used by the winning plan:

**db.restaurants.find({ name: "Glorious Food" })**

**Without indexing:**  


**With indexing:**



2. Drop index from task 4.1



3. Create an index to make this query **covered** and provide proof (from explain() or Compass UI) that it is indeed covered:

**db.restaurants.find({ restaurant\_id: "41098650" }, { \_id: 0, borough: 1 })**

**Without indexing:**

**Query:   
>db.restaurants.find({ restaurant\_id: "41098650" }, { \_id: 0, borough:1}).explain()**

****

**With indexing:**

**Creating index:  
>db.restaurants.createIndex({restaurant\_id:1, borough:1})**

**Query:  
>db.restaurants.find({ restaurant\_id: "41098650" }, { \_id: 0, borough:1}).explain()**

****

**4.** Create a **partial** index on cuisine field which will be used only when filtering on borough equal to “Staten Island”:

**db.restaurants.find({ borough: "Staten Island", cuisine: "American" }) –uses index**

**db.restaurants.find({ borough: "Staten Island", name: "Bagel Land" }) –does not use index**

**db.restaurants.find({ borough: "Queens", cuisine: "Pizza" }) –does not use index**

**Creating index:**

**>db.restaurants.createIndex({cuisine:1}, { partialFilterExpression: {borough: {$eq: "Staten Island"}}})**

****

**Query:  
> db.restaurants.find({ borough: "Staten Island", cuisine: "American" })  
**

**Query:   
> db.restaurants.find({ borough: "Staten Island", name: "Bagel Land"})**

****

**Query:**

**>db.restaurants.find({ borough: "Queens", cuisine: "Pizza"})  
**

5 Create an index to make query from task 3.4 **covered** and provide proof (from explain() or Compass UI) that it is indeed covered

*(What are the names of the restaurants which have a grade at index 8 with score less then 7? Use projection to include only names without \_id. )*

**Create index:**  
> db.restaurants.createIndex({"grades.8.score": 1}, {sparse: true})

**Query:**  
> db.restaurants.find({"grades.8.score": {$lt:7}}, {\_id:0, name:1}).explain()

**