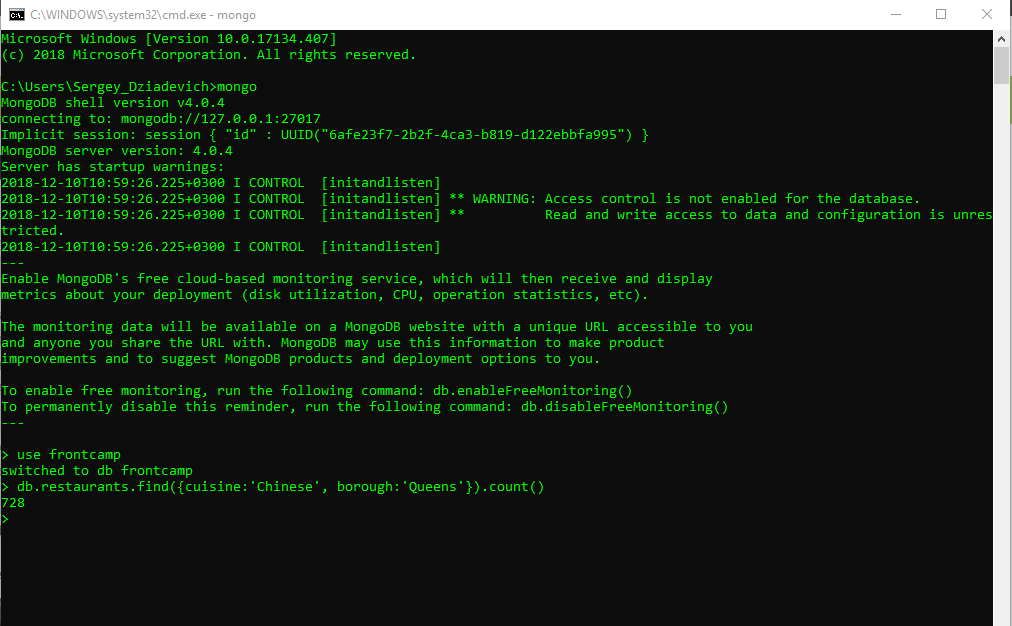
**3. Querying Restaurants Collection**

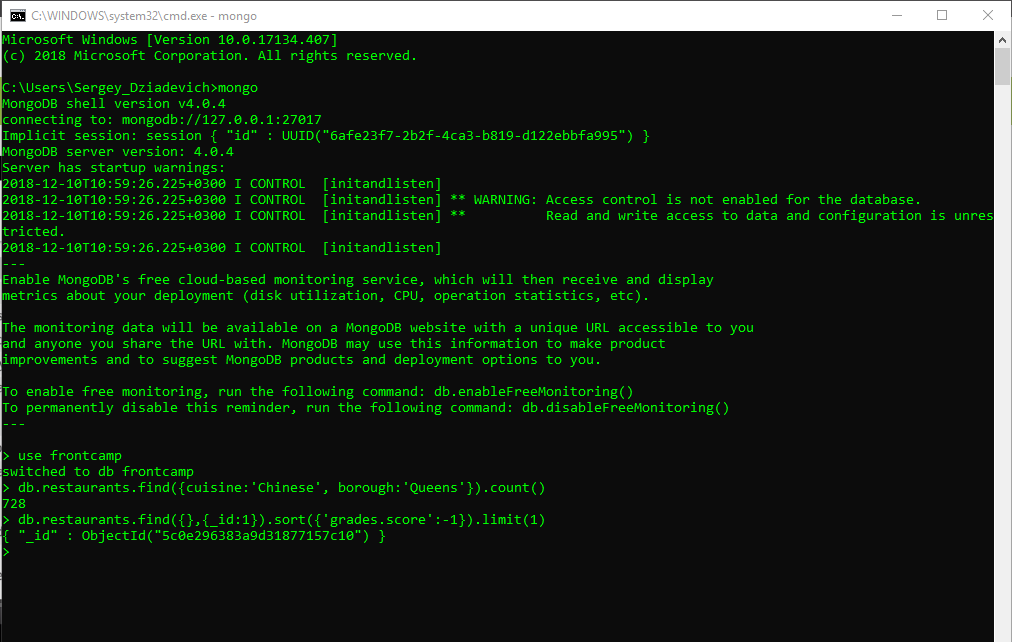
**1. How many “Chinese” (cuisine) restaurants are in “Queens” (borough)?**

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

**728** 

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

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

**{ "\_id" : ObjectId("5c0e296383a9d31877157c10") }**

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

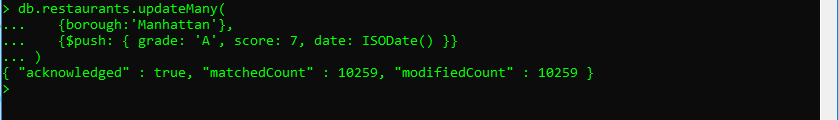
db.restaurants.updateMany(

{borough:'Manhattan'},

{$push: { grade: 'A', score: 7, date: ISODate() }}

)

**{ "acknowledged" : true, "matchedCount" : 10259, "modifiedCount" : 10259 }**

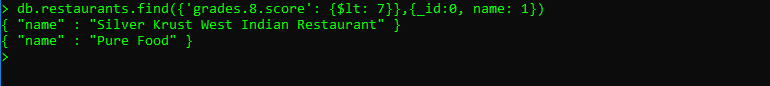
****

**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.**

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

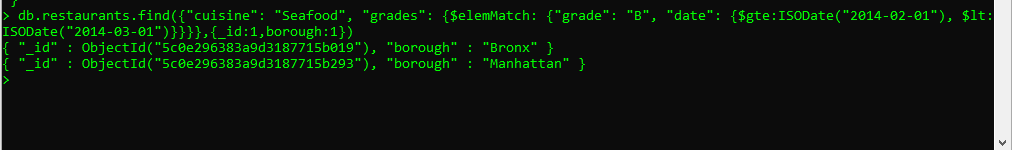
{ "name" : "Silver Krust West Indian Restaurant" }

{ "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.**

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

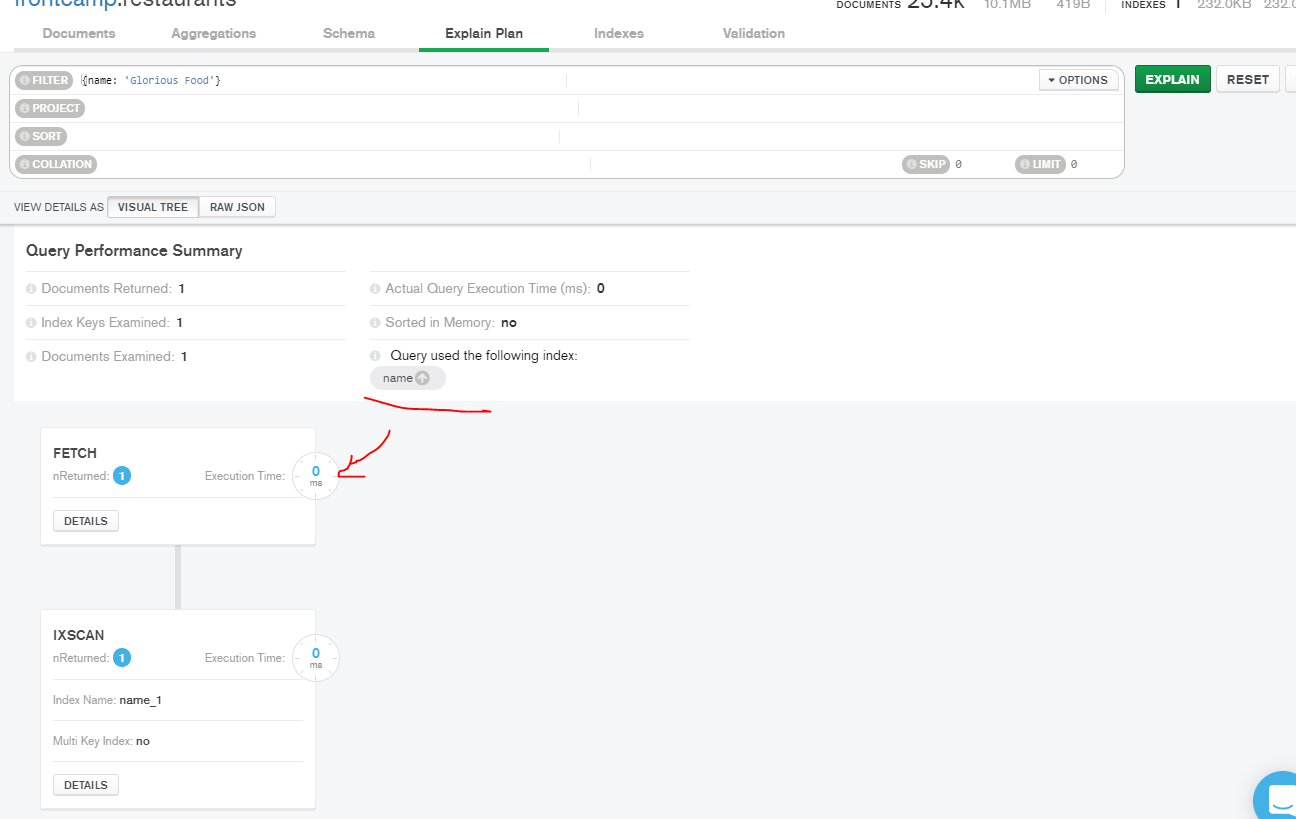
**{ "\_id" : ObjectId("5c0e296383a9d3187715b019"), "borough" : "Bronx" }  
{ "\_id" : ObjectId("5c0e296383a9d3187715b293"), "borough" : "Manhattan" }**

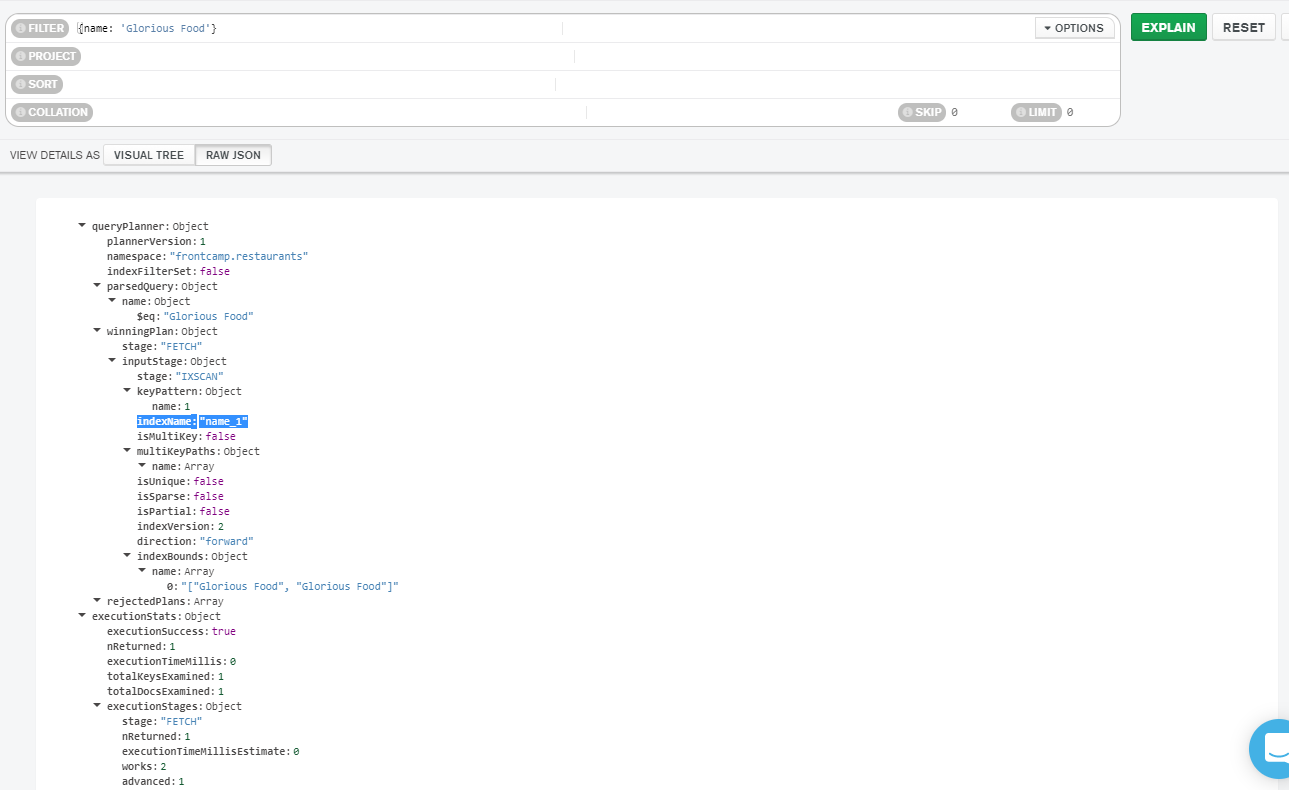
**4. Indexing Restaurants Collection (Note: you may use MongoDB Compass for this task)**

**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:**

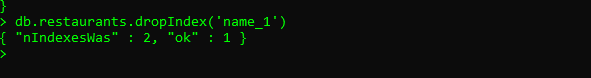
db.restaurants.createIndex({name:1})

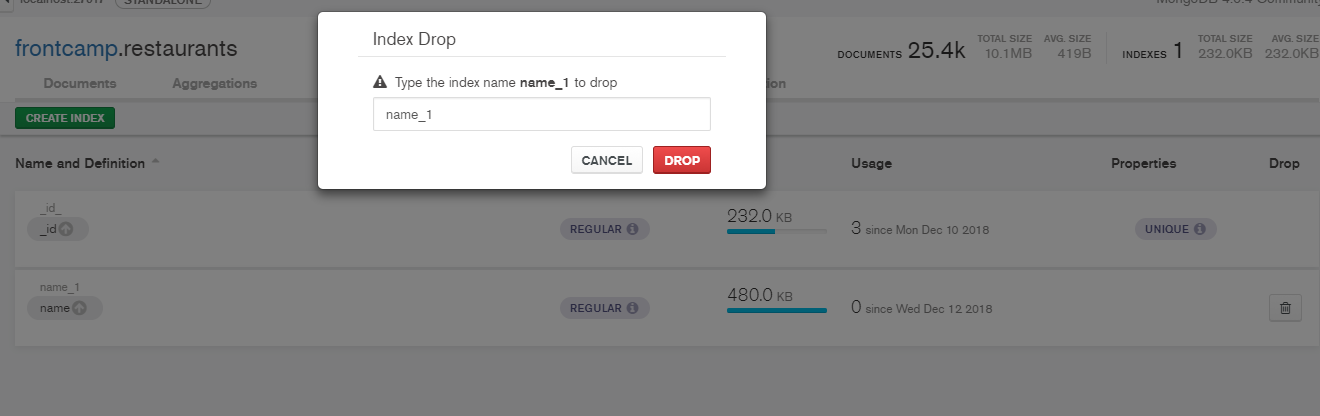
****

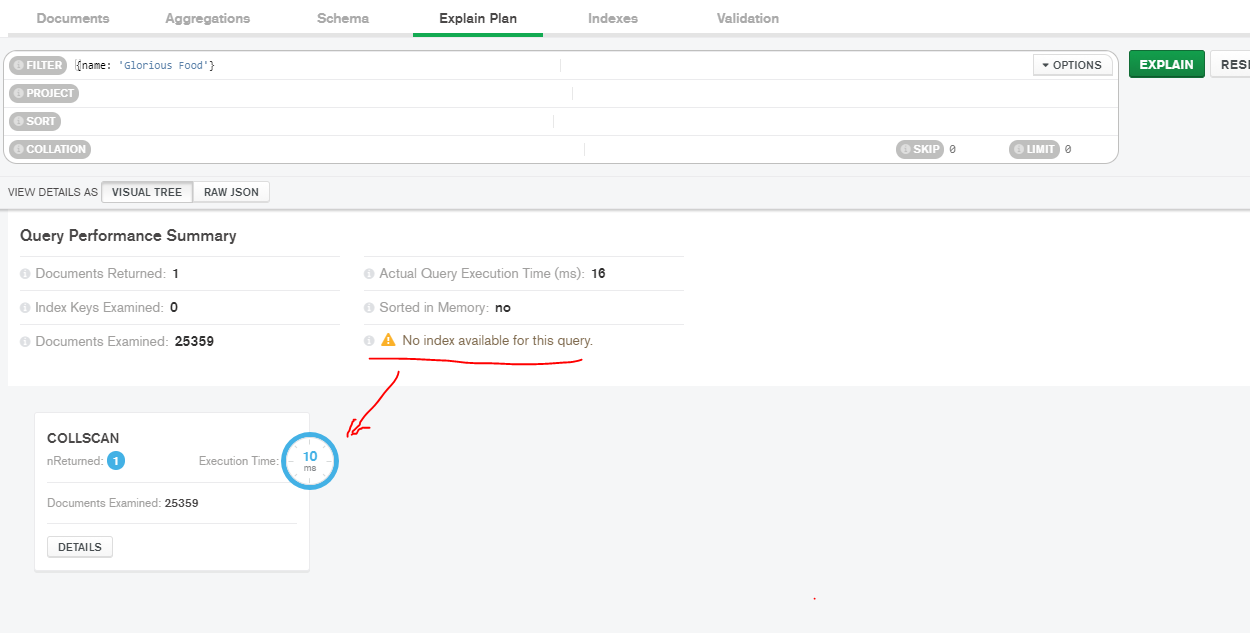
****

****

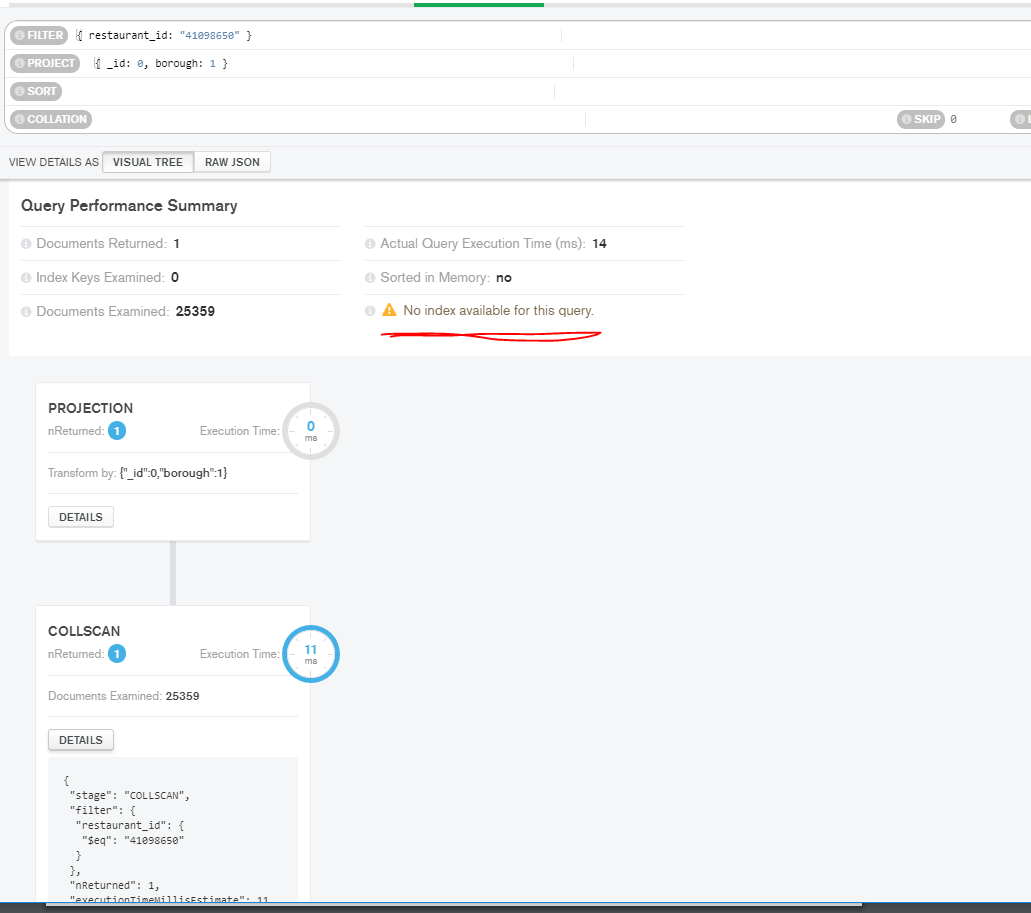
**2. Drop index from task 4.1**

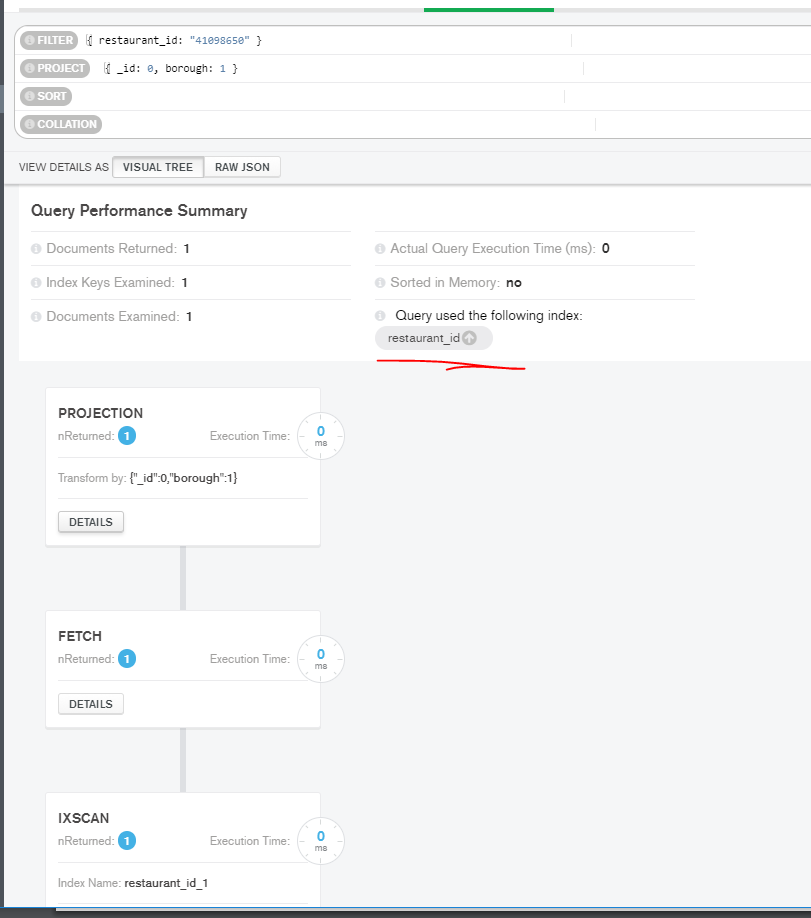
db.restaurants.dropIndex('name\_1')

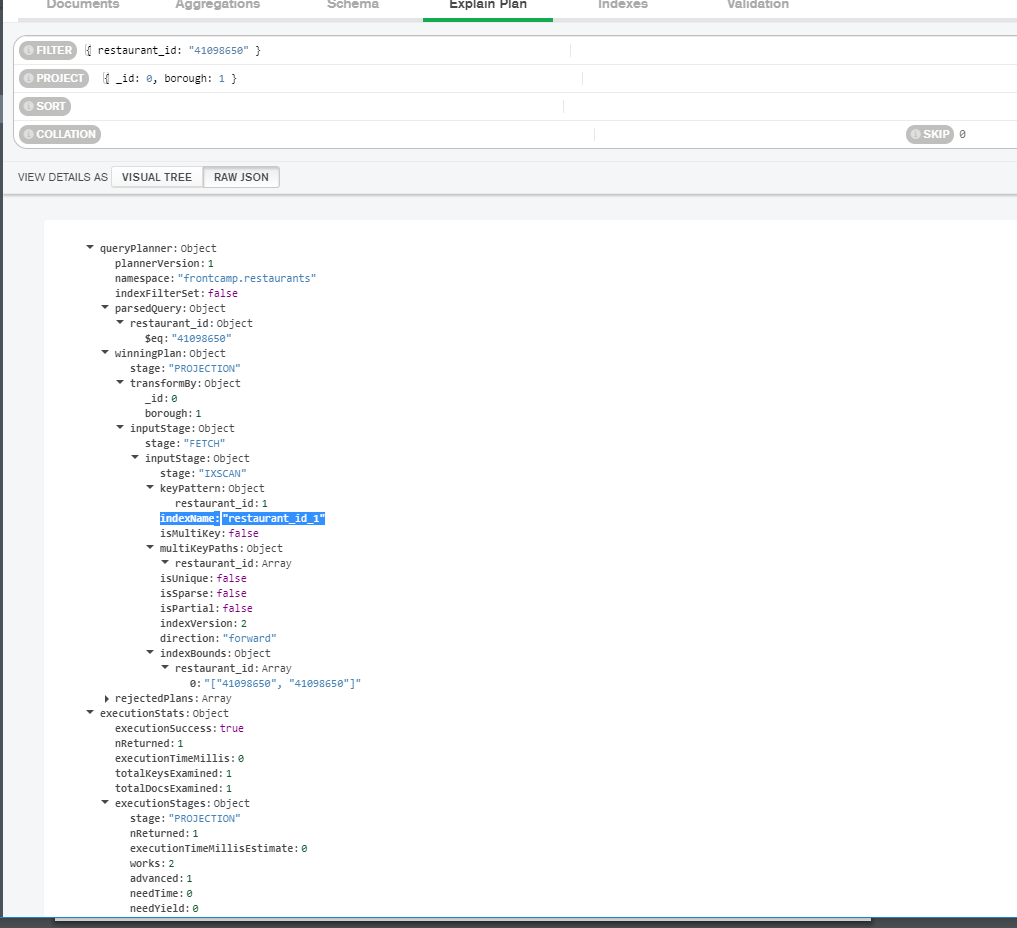
****

****

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

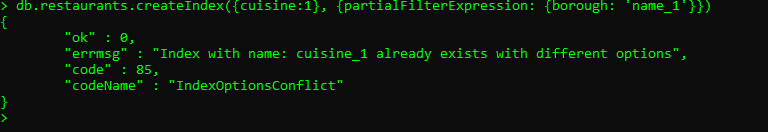
****

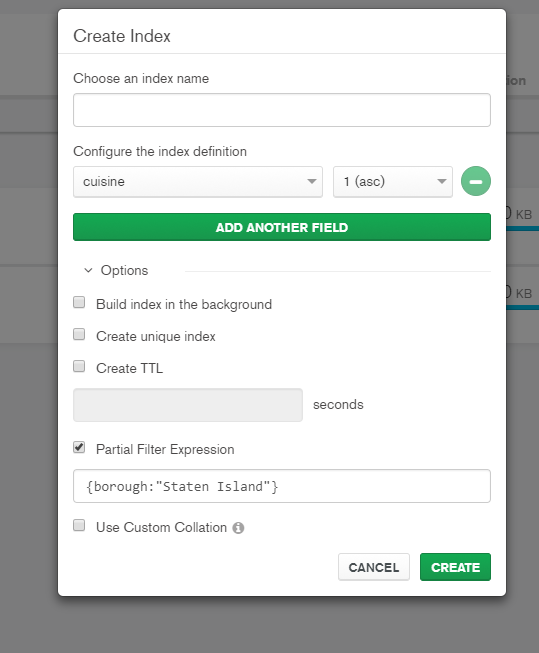
****

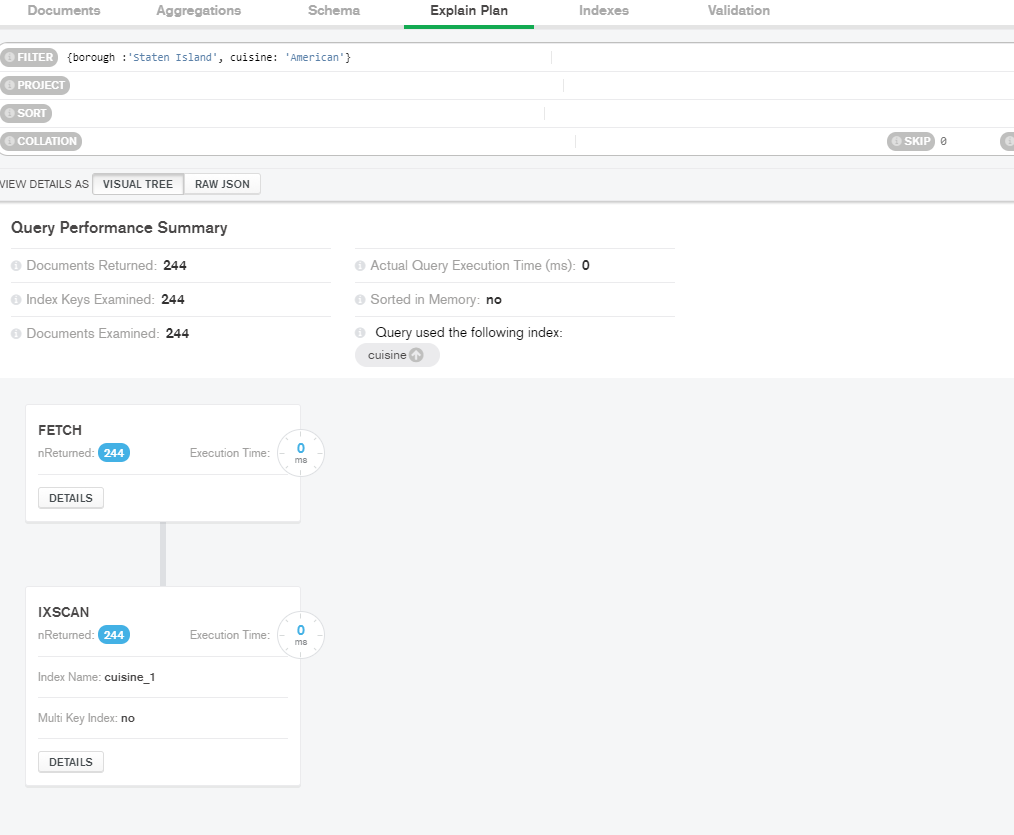
****

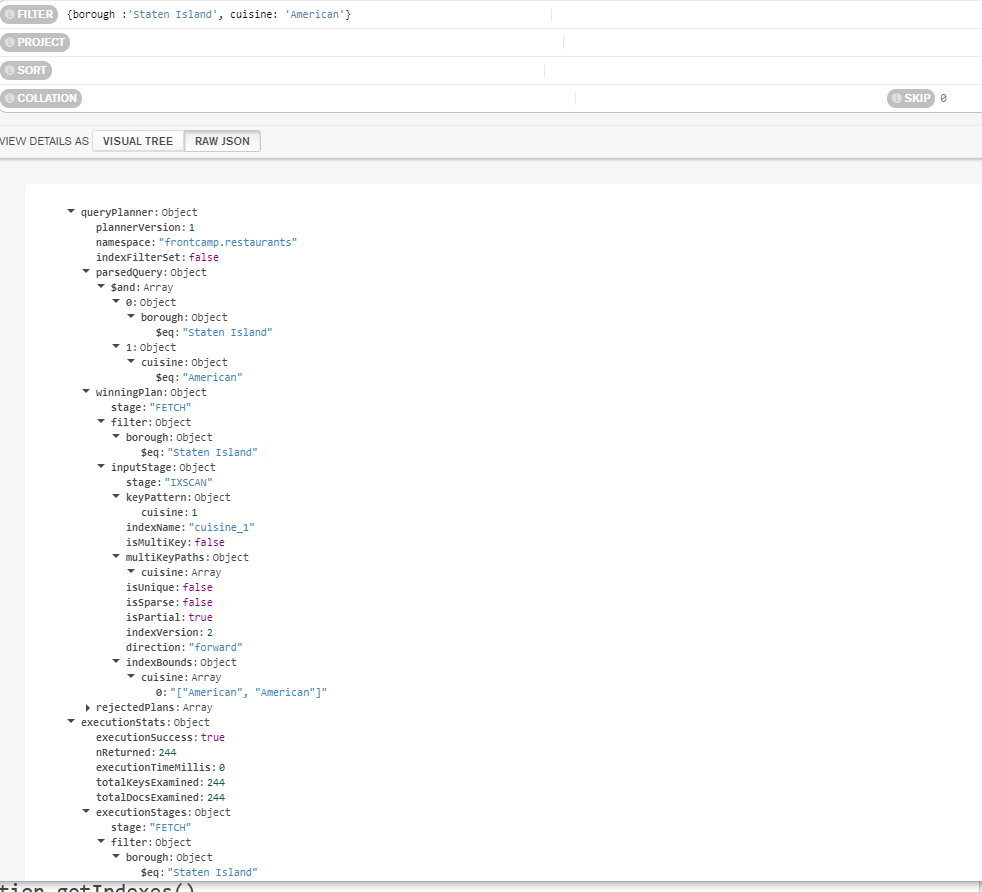
**4. Create a partial index on cuisine field which will be used only when filtering on borough equal to “Staten Island”:**

db.restaurants.createIndex({cuisine:1}, {partialFilterExpression: {borough: 'name\_1'}})



****

****

****