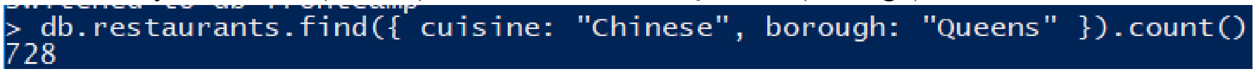
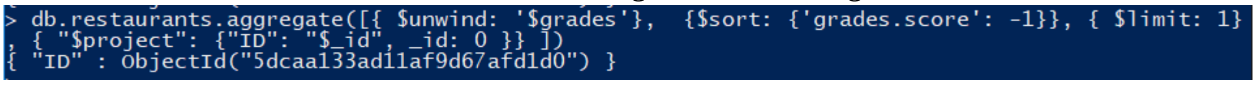
3. Querying Restaurants Collection

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



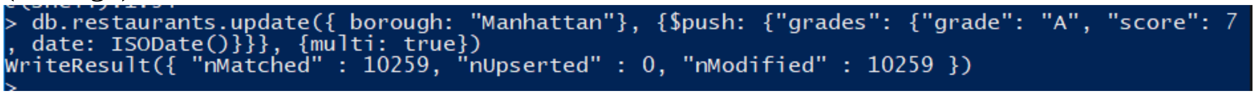
```
> db.restaurants.find({ cuisine: "Chinese", borough: "Queens" }).count()
728
```

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



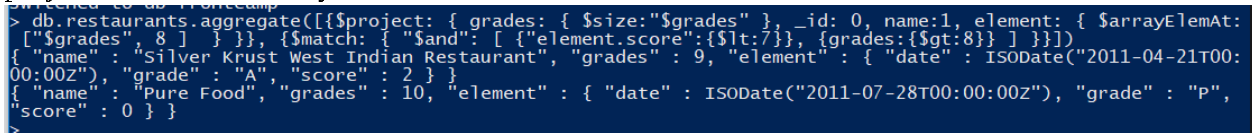
```
> db.restaurants.aggregate([{ $unwind: '$grades'},  {$sort: {'grades.score': -1}}, { $limit: 1}
, { "$project": {"ID": "$_id", _id: 0 }} ])
{ "ID" : ObjectId("5dcaa133ad11af9d67afd1d0") }
```

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



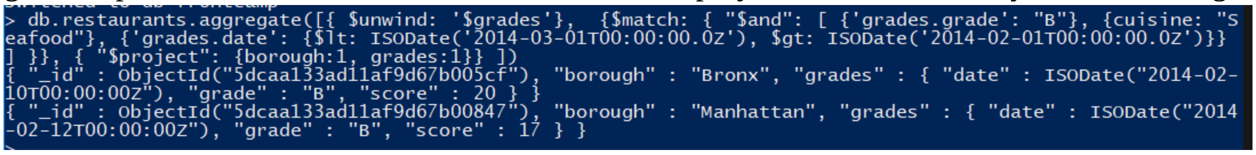
```
> db.restaurants.update({ borough: "Manhattan"}, {$push: {"grades": {"grade": "A", "score": 7
, date: ISODate()}}}, {multi: true})
WriteResult({ "nMatched" : 10259, "nUpserted" : 0, "nModified" : 10259 })
>
```

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



```
> db.restaurants.aggregate([{$project: { grades: { $size:"$grades" }, _id: 0, name:1, element: { $arrayElemAt:
 ["$grades", 8 ]  } }}, {$match: { "$and": [ {"element.score":{$lt:7}}, {grades:{$gt:8}} ] }}])
{ "name" : "Silver Krust West Indian Restaurant", "grades" : 9, "element" : { "date" : ISODate("2011-04-21T00:
00:00Z"), "grade" : "A", "score" : 2 } }
{ "name" : "Pure Food", "grades" : 10, "element" : { "date" : ISODate("2011-07-28T00:00:00Z"), "grade" : "P",
"score" : 0 } }
>
```

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.

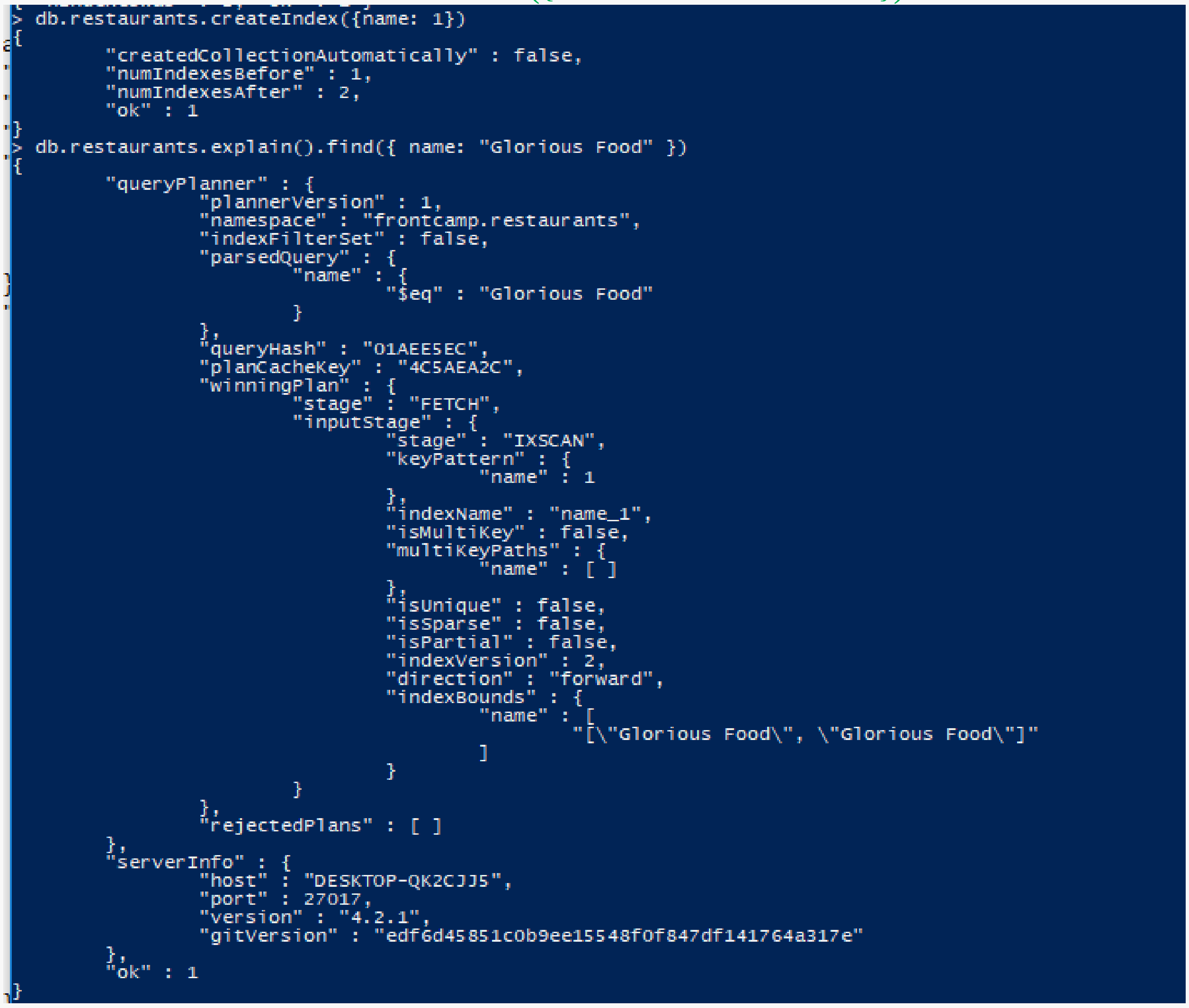


```
> db.restaurants.aggregate([{ $unwind: '$grades'},  {$match: { "$and": [ {'grades.grade': "B"}, {cuisine: "S
eafood"}, {'grades.date': {$lt: ISODate('2014-03-01T00:00:00.0Z'), $gt: ISODate('2014-02-01T00:00:00.0Z')}}
] }}, { "$project": {borough:1, grades:1}} ])
{ "_id" : ObjectId("5dcaa133ad11af9d67b005cf"), "borough" : "Bronx", "grades" : { "date" : ISODate("2014-02-
10T00:00:00Z"), "grade" : "B", "score" : 20 } }
{ "_id" : ObjectId("5dcaa133ad11af9d67b00847"), "borough" : "Manhattan", "grades" : { "date" : ISODate("2014
-02-12T00:00:00Z"), "grade" : "B", "score" : 17 } }
>
```

4. Indexing Restaurants Collection

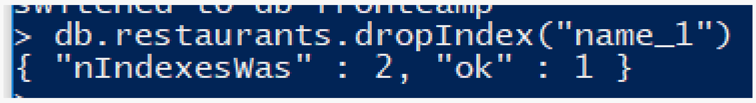
1.Create an index which will be used by this query and provide proof (from explain() or Compass UI) that the index is indeed used by the winning plan:

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



```
> db.restaurants.createIndex({name: 1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
> db.restaurants.explain().find({ name: "Glorious Food" })
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "frontcamp.restaurants",
                "indexFilterSet" : false,
                "parsedQuery" : {
                        "name" : {
                                "$eq" : "Glorious Food"
                        }
                },
                "queryHash" : "01AEE5EC",
                "planCacheKey" : "4C5AEA2C",
                "winningPlan" : {
                        "stage" : "FETCH",
                        "inputStage" : {
                                "stage" : "IXSCAN",
                                "keyPattern" : {
                                        "name" : 1
                                },
                                "indexName" : "name_1",
                                "isMultiKey" : false,
                                "multiKeyPaths" : {
                                        "name" : [ ]
                                },
                                "isUnique" : false,
                                "isSparse" : false,
                                "isPartial" : false,
                                "indexVersion" : 2,
                                "direction" : "forward",
                                "indexBounds" : {
                                        "name" : [
                                                "[\"Glorious Food\", \"Glorious Food\"]"
                                        ]
                                }
                        }
                },
                "rejectedPlans" : [ ]
        },
        "serverInfo" : {
                "host" : "DESKTOP-QK2CJJ5",
                "port" : 27017,
                "version" : "4.2.1",
                "gitVersion" : "edf6d45851c0b9ee15548f0f847df141764a317e"
        },
        "ok" : 1
}
```

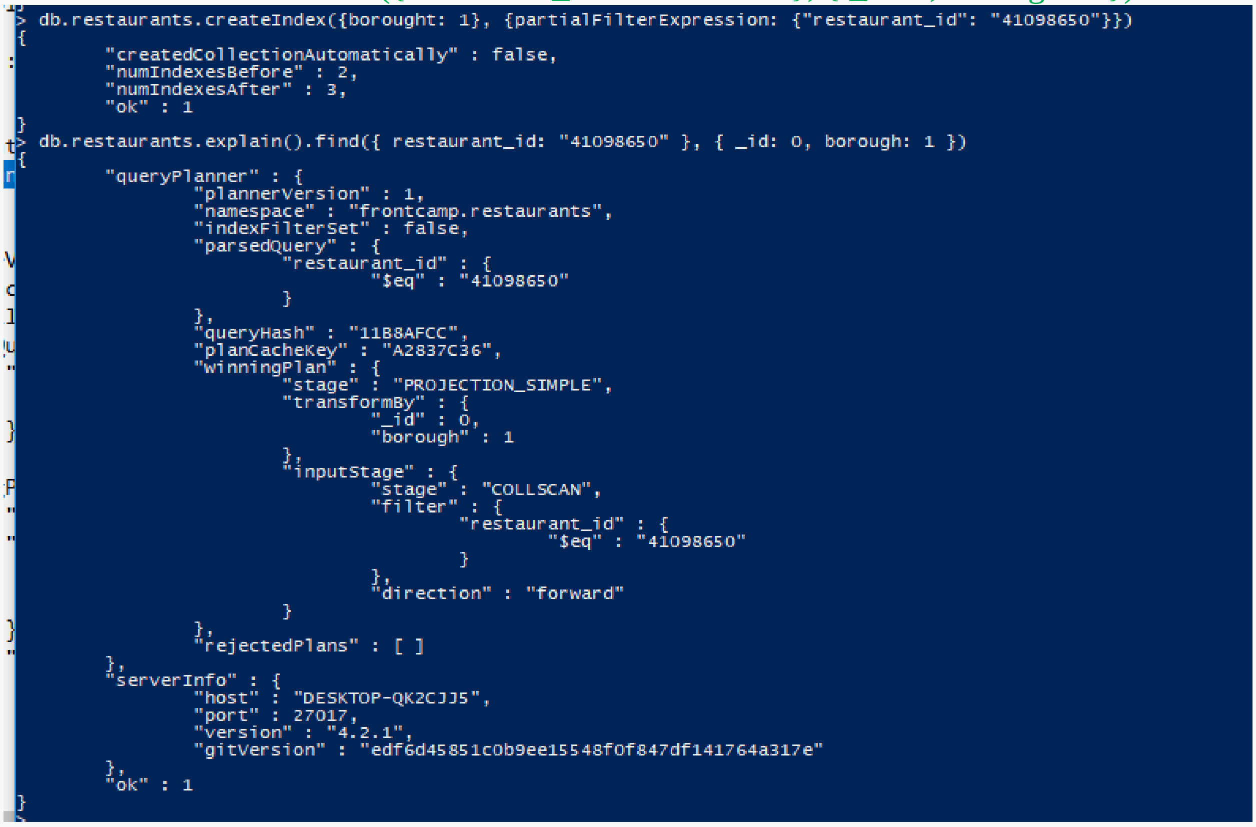
2.Drop index from task 4.1.



```
> db.restaurants.dropIndex("name_1")
{ "nIndexesWas" : 2, "ok" : 1 }
```

3.Create an index to make this query coveredand provide proof (from explain() or CompassUI)that it is indeed covered:

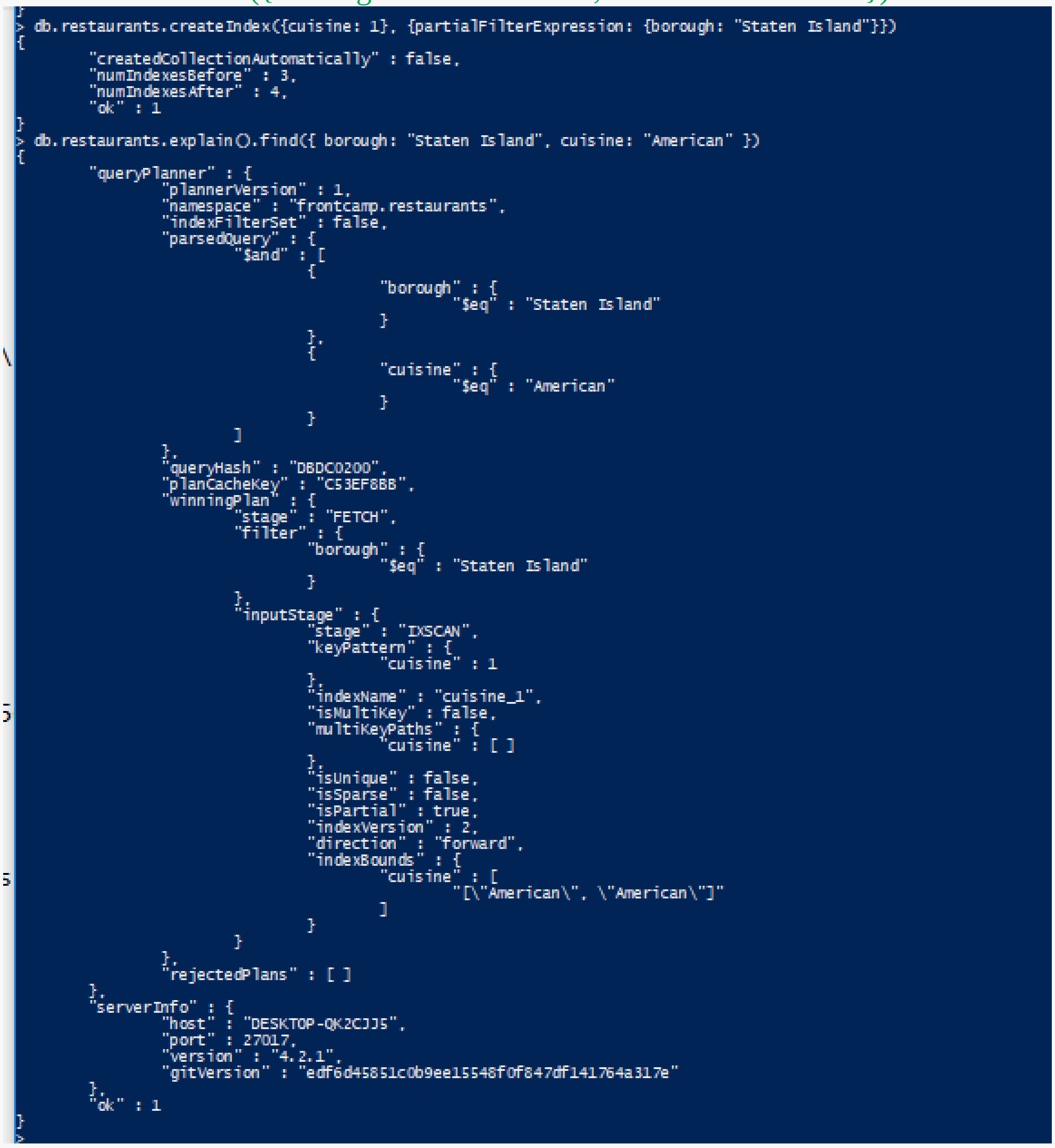
db.restaurants.find({ restaurant_id: "41098650" }, { _id: 0, borough: 1 })



```
> db.restaurants.createIndex({borought: 1}, {partialFilterExpression: {"restaurant_id": "41098650"}})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 2,
        "numIndexesAfter" : 3,
        "ok" : 1
}
> db.restaurants.explain().find({ restaurant_id: "41098650" }, { _id: 0, borough: 1 })
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "frontcamp.restaurants",
                "indexFilterSet" : false,
                "parsedQuery" : {
                        "restaurant_id" : {
                                "$eq" : "41098650"
                        }
                },
                "queryHash" : "11B8AFCC",
                "planCacheKey" : "A2837C36",
                "winningPlan" : {
                        "stage" : "PROJECTION_SIMPLE",
                        "transformBy" : {
                                "_id" : 0,
                                "borough" : 1
                        },
                        "inputStage" : {
                                "stage" : "COLLSCAN",
                                "filter" : {
                                        "restaurant_id" : {
                                                "$eq" : "41098650"
                                        }
                                },
                                "direction" : "forward"
                        }
                },
                "rejectedPlans" : [ ]
        },
        "serverInfo" : {
                "host" : "DESKTOP-QK2CJJ5",
                "port" : 27017,
                "version" : "4.2.1",
                "gitVersion" : "edf6d45851c0b9ee15548f0f847df141764a317e"
        },
        "ok" : 1
}
```

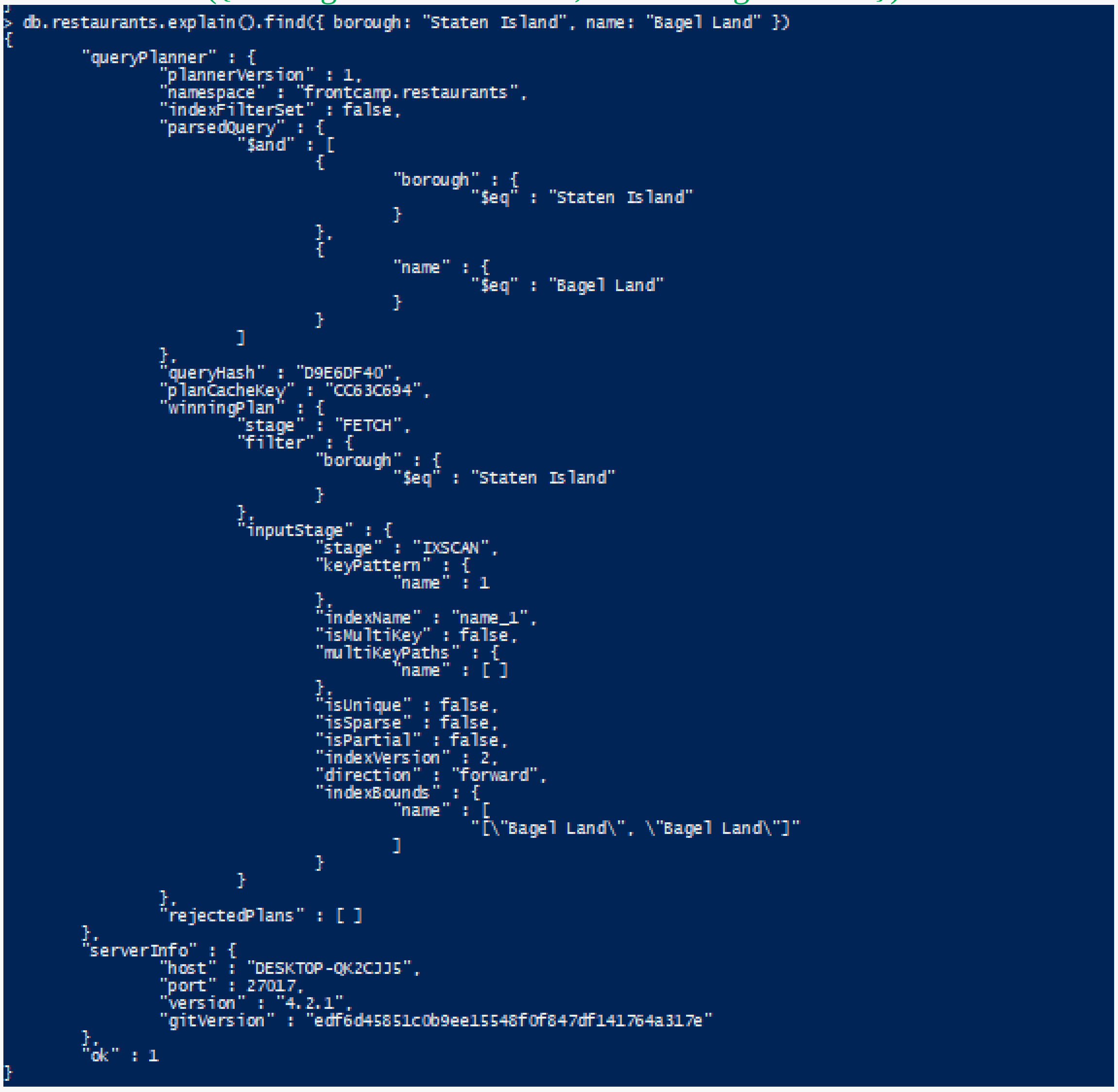
4.Create apartialindexon cuisinefieldwhich will be usedonly when filtering on borough equal to “Staten Island”:

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



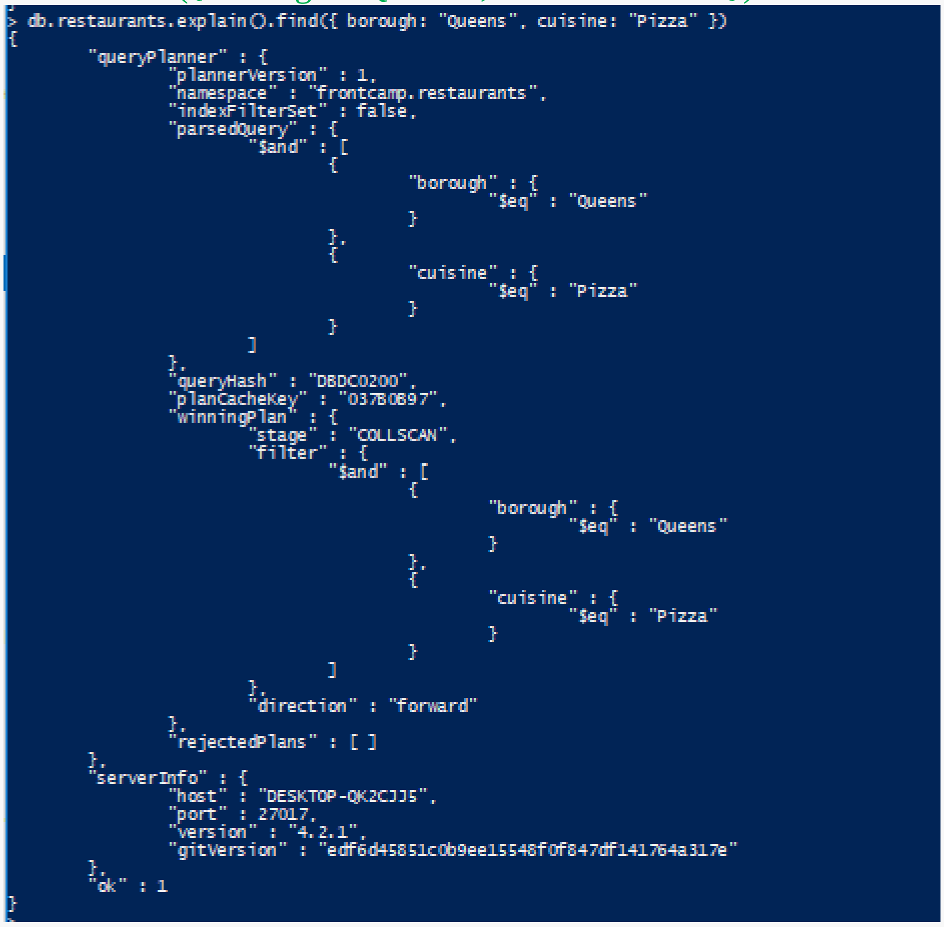
```
> db.restaurants.createIndex({cuisine: 1}, {partialFilterExpression: {borough: "Staten Island"}})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 3,
        "numIndexesAfter" : 4,
        "ok" : 1
}
> db.restaurants.explain().find({ borough: "Staten Island", cuisine: "American" })
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "frontcamp.restaurants",
                "indexFilterSet" : false,
                "parsedQuery" : {
                        "$and" : [
                                {
                                        "borough" : {
                                                "$eq" : "Staten Island"
                                        }
                                },
                                {
                                        "cuisine" : {
                                                "$eq" : "American"
                                        }
                                }
                        ]
                },
                "queryHash" : "DBDC0200",
                "planCacheKey" : "C53EF8BB",
                "winningPlan" : {
                        "stage" : "FETCH",
                        "filter" : {
                                "borough" : {
                                        "$eq" : "Staten Island"
                                }
                        },
                        "inputStage" : {
                                "stage" : "IXSCAN",
                                "keyPattern" : {
                                        "cuisine" : 1
                                },
                                "indexName" : "cuisine_1",
                                "isMultiKey" : false,
                                "multiKeyPaths" : {
                                        "cuisine" : [ ]
                                },
                                "isUnique" : false,
                                "isSparse" : false,
                                "isPartial" : true,
                                "indexVersion" : 2,
                                "direction" : "forward",
                                "indexBounds" : {
                                        "cuisine" : [
                                                "[\"American\", \"American\"]"
                                        ]
                                }
                        }
                },
                "rejectedPlans" : [ ]
        },
        "serverInfo" : {
                "host" : "DESKTOP-QK2CJJ5",
                "port" : 27017,
                "version" : "4.2.1",
                "gitVersion" : "edf6d45851c0b9ee15548f0f847df141764a317e"
        },
        "ok" : 1
}
>
```

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



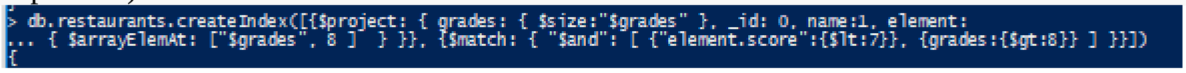
```
> db.restaurants.explain().find({ borough: "Staten Island", name: "Bagel Land" })
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "frontcamp.restaurants",
                "indexFilterSet" : false,
                "parsedQuery" : {
                        "$and" : [
                                {
                                        "borough" : {
                                                "$eq" : "Staten Island"
                                        }
                                },
                                {
                                        "name" : {
                                                "$eq" : "Bagel Land"
                                        }
                                }
                        ]
                },
                "queryHash" : "D9E6DF40",
                "planCacheKey" : "CC63C694",
                "winningPlan" : {
                        "stage" : "FETCH",
                        "filter" : {
                                "borough" : {
                                        "$eq" : "Staten Island"
                                }
                        },
                        "inputStage" : {
                                "stage" : "IXSCAN",
                                "keyPattern" : {
                                        "name" : 1
                                },
                                "indexName" : "name_1",
                                "isMultiKey" : false,
                                "multiKeyPaths" : {
                                        "name" : [ ]
                                },
                                "isUnique" : false,
                                "isSparse" : false,
                                "isPartial" : false,
                                "indexVersion" : 2,
                                "direction" : "forward",
                                "indexBounds" : {
                                        "name" : [
                                                "[\"Bagel Land\", \"Bagel Land\"]"
                                        ]
                                }
                        }
                },
                "rejectedPlans" : [ ]
        },
        "serverInfo" : {
                "host" : "DESKTOP-QK2CJJ5",
                "port" : 27017,
                "version" : "4.2.1",
                "gitVersion" : "edf6d45851c0b9ee15548f0f847df141764a317e"
        },
        "ok" : 1
}
```

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



```
> db.restaurants.explain().find({ borough: "Queens", cuisine: "Pizza" })
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "frontcamp.restaurants",
                "indexFilterSet" : false,
                "parsedQuery" : {
                        "$and" : [
                                {
                                        "borough" : {
                                                "$eq" : "Queens"
                                        }
                                },
                                {
                                        "cuisine" : {
                                                "$eq" : "Pizza"
                                        }
                                }
                        ]
                },
                "queryHash" : "DBDC0200",
                "planCacheKey" : "037B0B97",
                "winningPlan" : {
                        "stage" : "COLLSCAN",
                        "filter" : {
                                "$and" : [
                                        {
                                                "borough" : {
                                                        "$eq" : "Queens"
                                                }
                                        },
                                        {
                                                "cuisine" : {
                                                        "$eq" : "Pizza"
                                                }
                                        }
                                ]
                        },
                        "direction" : "forward"
                },
                "rejectedPlans" : [ ]
        },
        "serverInfo" : {
                "host" : "DESKTOP-QK2CJJ5",
                "port" : 27017,
                "version" : "4.2.1",
                "gitVersion" : "edf6d45851c0b9ee15548f0f847df141764a317e"
        },
        "ok" : 1
}
```

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.



```
> db.restaurants.createIndex([{$project: { grades: { $size:"$grades" }, _id: 0, name:1, element:
... { $arrayElemAt: ["$grades", 8 ]  } }}, {$match: { "$and": [ {"element.score":{$lt:7}}, {grades:{$gt:8}} ] }}])
{
```