

**SCHOOL OF COMPUTER SCIENCE ENGINEERING**

**AND INFORMATION SYSTEMS**

**FALL SEMESTER 2024-2025**

**PMCA503P – DATABASE SYSTEMS LAB**

**MONGO-DB CYCLESHEET**

**SUBMITTED ON: 09 – NOV - 2024**

**SUBMITTED BY-**

**AKASH KUMAR BANIK**

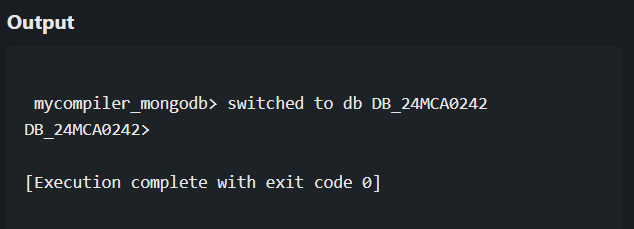
**PROGRAM: MCA**

**REGISTER No.: 24MCA0242**

**CYCLE SHEET-MONGODB AND APPLICATION DEVELOPMENT**

**Create the restaurants collection in your DB [DB\_regno]**

use DB\_24MCA0242



**Import the json file to your collection [restaurants.json]**

**Restaurants.json file:**

[

{

"restaurant\_id": "100001",

"name": "Central Cafe",

"borough": "Manhattan",

"cuisine": "Italian",

"ratings": [

{ "date": "2023-05-01T00:00:00Z", "grade": "A", "score": 95 }

]

},

{

"restaurant\_id": "100002",

"name": "Green Garden",

"borough": "Brooklyn",

"cuisine": "Vegetarian",

"ratings": [

{ "date": "2023-06-10T00:00:00Z", "grade": "B", "score": 82 }

]

},

{

"restaurant\_id": "100003",

"name": "Spicy House",

"borough": "Queens",

"cuisine": "Mexican",

"ratings": [

{ "date": "2023-07-15T00:00:00Z", "grade": "A", "score": 90 }

]

},

{

"restaurant\_id": "100004",

"name": "Ocean View",

"borough": "Bronx",

"cuisine": "Seafood",

"ratings": [

{ "date": "2023-08-21T00:00:00Z", "grade": "A", "score": 88 }

]

},

{

"restaurant\_id": "100005",

"name": "Sunshine Diner",

"borough": "Staten Island",

"cuisine": "American",

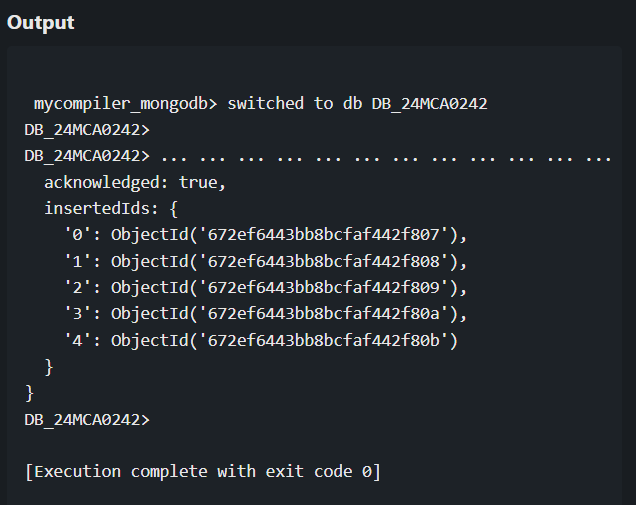
"ratings": [

{ "date": "2023-09-30T00:00:00Z", "grade": "C", "score": 72 }

]

}

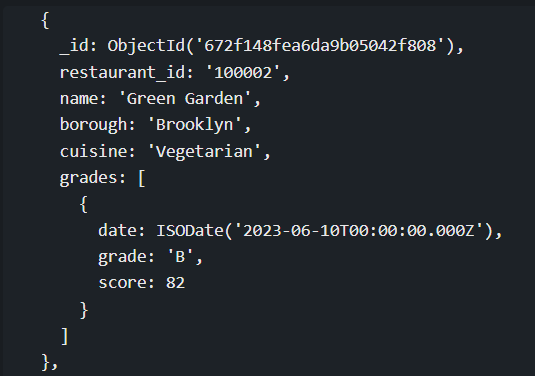
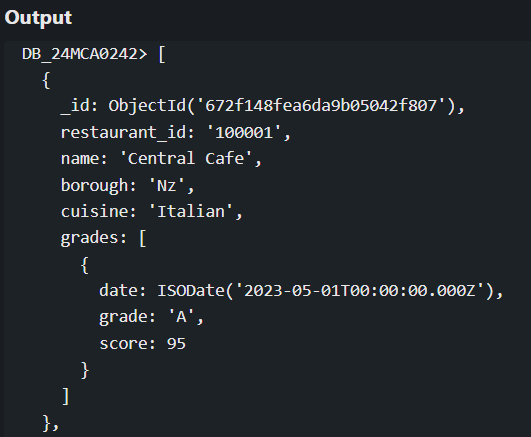
]

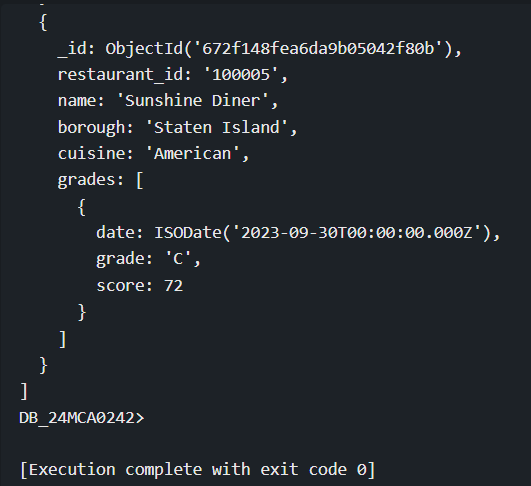
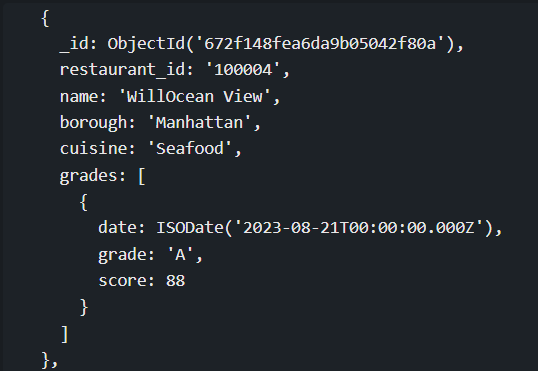
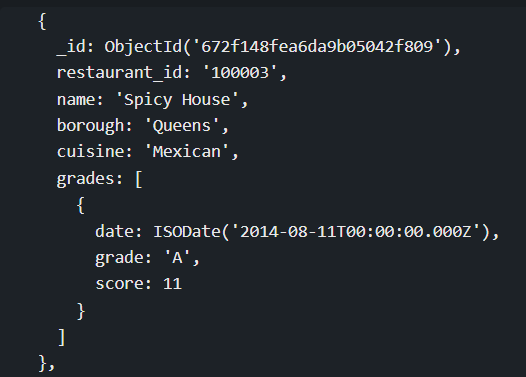


**Write mongodb queries to**

1. Display all the documents in the collection restaurants.

db.restaurants.find({})

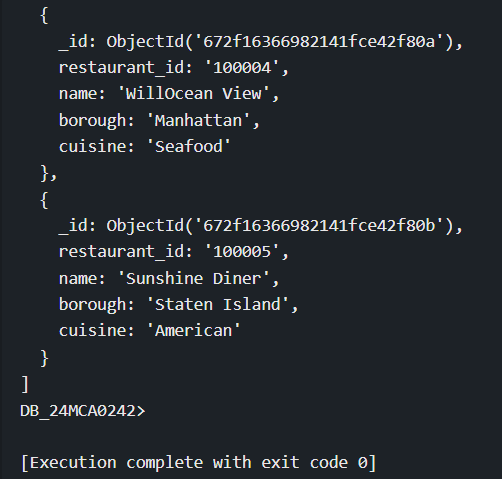
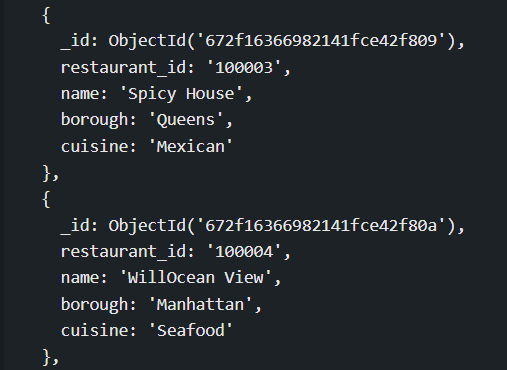




1. Display the fields restaurant\_id, name, borough and cuisine for all the documents in the collection restaurant.

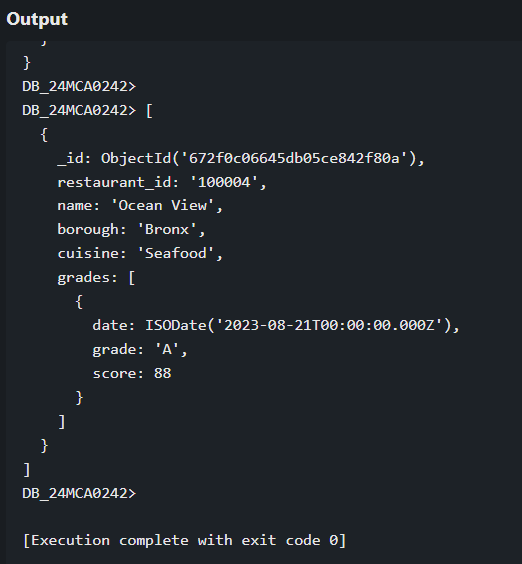
db.restaurants.find({}, { restaurant\_id: 1, name: 1, borough: 1, cuisine: 1 })





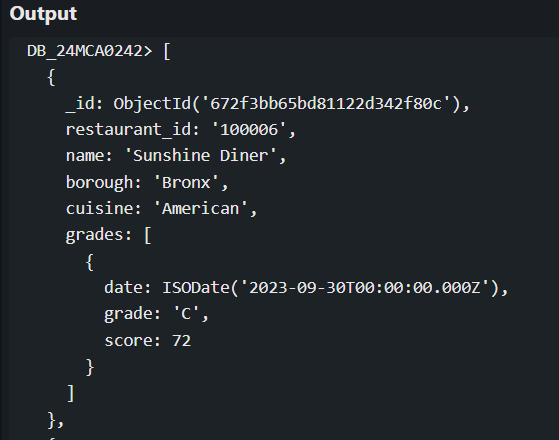
1. Display all the restaurant which is in the borough Bronx.

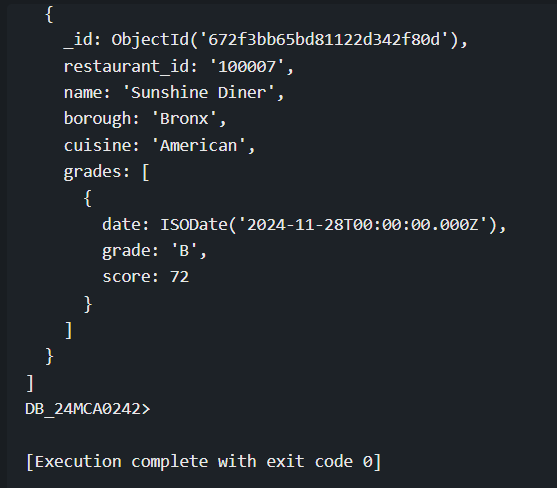
db.restaurants.find({ borough: "Bronx" })



1. Display the next 5 restaurants after skipping first 5 which are in the borough Bronx.

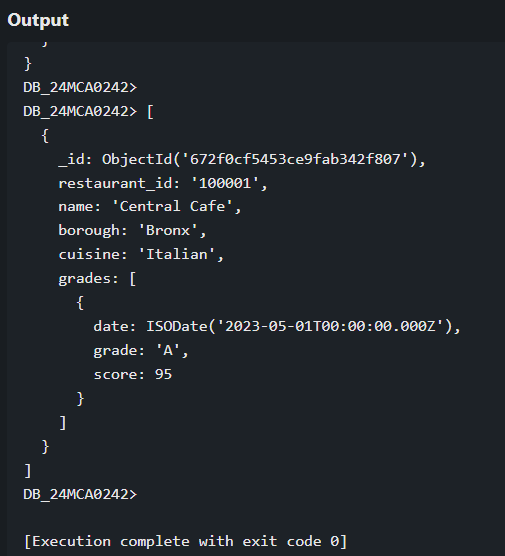
db.restaurants.find({ borough: "Bronx" }).skip(5).limit(5)





1. Find the restaurants who achieved a score more than 90.

db.restaurants.find({ "grades.score": { $gt: 90 } })



1. Find the restaurants that do not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than -65.754168.

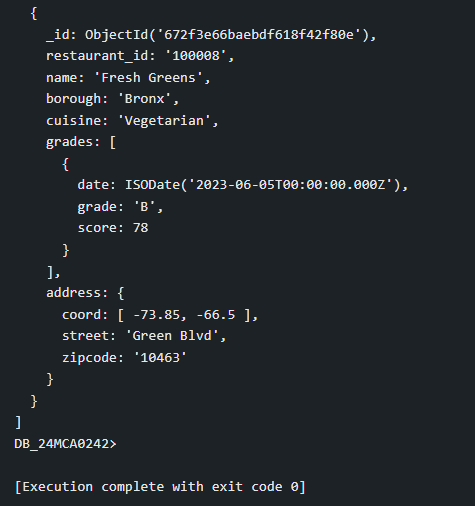
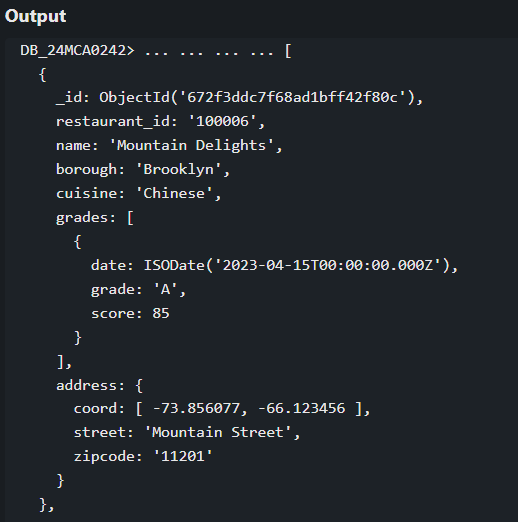
db.restaurants.find({

cuisine: { $ne: "American" },

"grades.score": { $gt: 70 },

"address.coord.1": { $lt: -65.754168 }

})



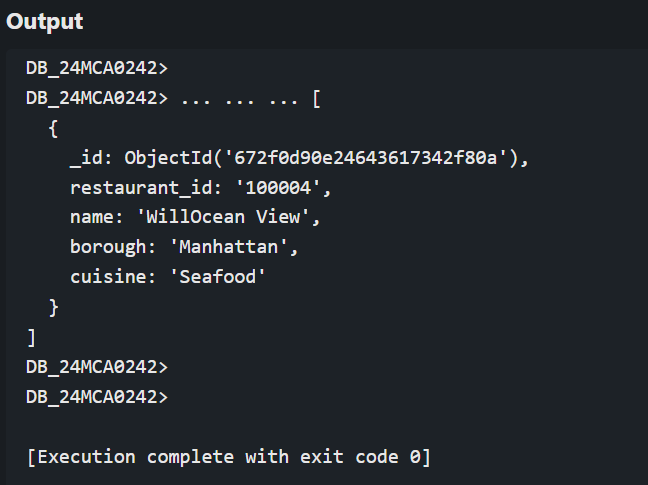
1. Find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Wil' as first three letters for its name.

db.restaurants.find(

{ name: { $regex: /^Wil/ } },

{ restaurant\_id: 1, name: 1, borough: 1, cuisine: 1 }

)



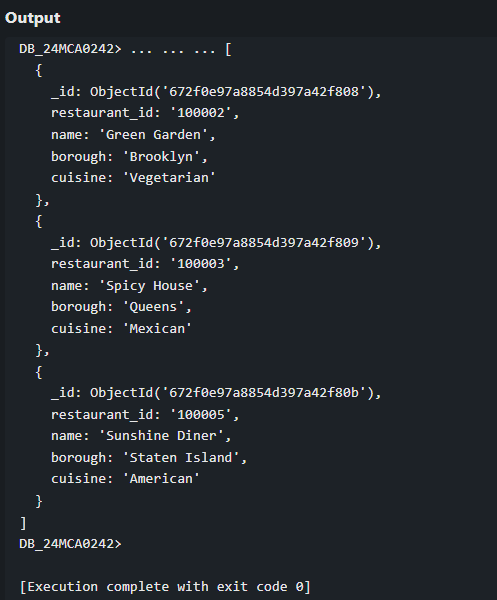
1. Find the restaurant Id, name, borough and cuisine for those restaurants which belong to the borough Staten Island or Queens or Bronx or Brooklyn.

db.restaurants.find(

{ borough: { $in: ["Staten Island", "Queens", "Bronx", "Brooklyn"] } },

{ restaurant\_id: 1, name: 1, borough: 1, cuisine: 1 }

)



1. Find the restaurant Id, name, and grades for those restaurants which achieved a grade of "A" and scored 11 on an isodate "2014-08-11" among many of survey dates.

db.restaurants.find({

grades: {

$elemMatch: {

grade: "A",

score: 11,

date: ISODate("2014-08-11T00:00:00Z")

}

}

}, { restaurant\_id: 1, name: 1, grades: 1 })



1. Arrange the name of the restaurants in descending along with all the columns.

db.restaurants.find({}).sort({ name: -1 })

