Assignment 3:

A screenshot of a computer

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

show dbs

mangodb\_practice 72.00 KiB

mongodb\_practice 8.00 KiB

population 4.55 MiB

restaurants 1.42 MiB

test 160.00 KiB

admin 332.00 KiB

local 3.80 GiB

use restaurant

|  |
| --- |
| **'switched to db restaurant'**  **Exercise Questions:-**   1. db.addresses.find() 2. db.addresses.aggregate([{$project:{restaurant\_id:1,name:1,borough:1,cuisine:1}}]) 3. db.addresses.aggregate([{$project:{restaurant\_id:1,name:1,borough:1,cuisine:1,\_id:0}}]) 4. db.addresses.aggregate([{$project:{restaurant\_id:1,name:1,borough:1,"address.zipcode":1,\_id:0}}]) 5. db.addresses.find({borough:"Bronx"}).limit(5) 6. db.addresses.find({borough:"Bronx"}) 7. db.addresses.find({borough:"Bronx"}).skip(5).limit(5) 8. db.addresses.find({"grades.score":{$gt:90}}) 9. db.addresses.find({$and:[{"grades.score":{$gt:80}},{"grades.score":{$lt:100}}]}) 10. db.addresses.find({"address.coord.0":{$lt: -95.754168}}) 11. db.addresses.find({$and:[{cuisine:{$ne:"American"}}, 12. db.addresses.find({$and:[{cuisine:{$ne:"American"}},   {"grades.score":{$gt:70}},{"address.coord.1":{$lt: 65.754168}}]})   1. db.addresses.find({$and:[{cuisine:{$ne:"American"}},   {"grades.grade":"A"},{borough:{$ne:"Brooklyn"}}]}).sort({cuisine:-1})   1. db.addresses.find({name:/^Wil/},{restaurant\_id:1,name:1,borough:1,cuisine:1}) 2. db.addresses.find({name:{$regex:"Reg"}},{restaurant\_id:1,name:1,borough:1,cuisine:1}) 3. db.addresses.find({$and:[{borough:"Bronx"},{$or:[{cuisine:"American"},   {cuisine:"Chinese"}]}]})   1. db.addresses.find({borough:{$in:["Staten Island","Queens","Bronx","Brooklyn"]}},{restaurant\_id:1,name:1,borough:1,cuisine:1}) 2. db.addresses.find({borough:{$nin:["StatenIsland","Queens","Bronx","Brooklyn"]}},   {restaurant\_id:1,name:1,borough:1,cuisine:1})   1. db.addresses.find({"grades.score":{$not:{$gt:10}}},{restaurant\_id:1,name:1,borough:1,cuisine:1}) 2. db.addresses.find({$or:[{$and:[{cuisine:{$ne:"American"}},{cuisine:{$ne:"Chinese"}}]},   {name:"/^Wil/"}]},{restaurent\_id:1,name:1,borough:1,cuisine:1})   1. db.addresses.find({$and:[{"grades.grade":"A"},{"grades.score":11},{"grades.date":ISODate("2014-08-11T00:00:00Z")}]},{restaurant\_id:1,name:1,grades:1}) 2. db.addresses.find({$and:[{"grades.1.grade":"A"},{"grades.1.score":9},   {"grades.1.date":ISODate("2014-0811T00:00:00Z")}]},{restaurant\_id:1,name:1,grades:1})   1. db.addresses.find().sort({name:1}) 2. db.addresses.find().sort({name:-1}) 3. db.addresses.find().sort({cuisine:1,borough:-1}) 4. db.addresses.find({"address.street":{$exists:true}}) |
|  |