MongoDB – Complex Queries

Exercise Questions:

1. Write a MongoDB query to display all the documents in the collection restaurants.

=> db.addresses.find()

2. Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine for all the documents in the collection restaurant.

=> db.addresses.aggregate({$project:{restaurant\_id:1,name:1,borough:1,cuisine:1}})

{ "\_id" : ObjectId("61f2711fc4002f57a303c2b3"), "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Morris Park Bake Shop", "restaurant\_id" : "30075445" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2b4"), "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "Wendy'S", "restaurant\_id" : "30112340" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2b5"), "borough" : "Manhattan", "cuisine" : "Irish", "name" : "Dj Reynolds Pub And Restaurant", "restaurant\_id" : "30191841" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2b6"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Riviera Caterer", "restaurant\_id" : "40356018" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2b7"), "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Tov Kosher Kitchen", "restaurant\_id" : "40356068" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2b8"), "borough" : "Queens", "cuisine" : "American ", "name" : "Brunos On The Boulevard", "restaurant\_id" : "40356151" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2b9"), "borough" : "Staten Island", "cuisine" : "Jewish/Kosher", "name" : "Kosher Island", "restaurant\_id" : "40356442" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2ba"), "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Wilken'S Fine Food", "restaurant\_id" : "40356483" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2bb"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Regina Caterers", "restaurant\_id" : "40356649" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2bc"), "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Taste The Tropics Ice Cream", "restaurant\_id" : "40356731" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2bd"), "borough" : "Bronx", "cuisine" : "American ", "name" : "Wild Asia", "restaurant\_id" : "40357217" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2be"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "C & C Catering Service", "restaurant\_id" : "40357437" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2bf"), "borough" : "Brooklyn", "cuisine" : "Chinese", "name" : "May May Kitchen", "restaurant\_id" : "40358429" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2c0"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "1 East 66Th Street Kitchen", "restaurant\_id" : "40359480" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2c1"), "borough" : "Brooklyn", "cuisine" : "Jewish/Kosher", "name" : "Seuda Foods", "restaurant\_id" : "40360045" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2c2"), "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant\_id" : "40360076" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2c3"), "borough" : "Queens", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant\_id" : "40361322" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2c4"), "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Nordic Delicacies", "restaurant\_id" : "40361390" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2c5"), "borough" : "Manhattan", "cuisine" : "American ", "name" : "Glorious Food", "restaurant\_id" : "40361521" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2c6"), "borough" : "Brooklyn", "cuisine" : "American ", "name" : "The Movable Feast", "restaurant\_id" : "40361606" }

Type "it" for more

3. Write a MongoDB query to display the fields restaurant\_id, name, borough and cuisine, but exclude the field \_id for all the documents in the collection restaurant.

=> db.addresses.aggregate({$project:{restaurant\_id:1,name:1,borough:1,cuisine:1,\_id:0}})

{ "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Morris Park Bake Shop", "restaurant\_id" : "30075445" }

{ "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "Wendy'S", "restaurant\_id" : "30112340" }

{ "borough" : "Manhattan", "cuisine" : "Irish", "name" : "Dj Reynolds Pub And Restaurant", "restaurant\_id" : "30191841" }

{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Riviera Caterer", "restaurant\_id" : "40356018" }

{ "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Tov Kosher Kitchen", "restaurant\_id" : "40356068" }

{ "borough" : "Queens", "cuisine" : "American ", "name" : "Brunos On The Boulevard", "restaurant\_id" : "40356151" }

{ "borough" : "Staten Island", "cuisine" : "Jewish/Kosher", "name" : "Kosher Island", "restaurant\_id" : "40356442" }

{ "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Wilken'S Fine Food", "restaurant\_id" : "40356483" }

{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Regina Caterers", "restaurant\_id" : "40356649" }

{ "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Taste The Tropics Ice Cream", "restaurant\_id" : "40356731" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Wild Asia", "restaurant\_id" : "40357217" }

{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "C & C Catering Service", "restaurant\_id" : "40357437" }

{ "borough" : "Brooklyn", "cuisine" : "Chinese", "name" : "May May Kitchen", "restaurant\_id" : "40358429" }

{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "1 East 66Th Street Kitchen", "restaurant\_id" : "40359480" }

{ "borough" : "Brooklyn", "cuisine" : "Jewish/Kosher", "name" : "Seuda Foods", "restaurant\_id" : "40360045" }

{ "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant\_id" : "40360076" }

{ "borough" : "Queens", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant\_id" : "40361322" }

{ "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Nordic Delicacies", "restaurant\_id" : "40361390" }

{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Glorious Food", "restaurant\_id" : "40361521" }

{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "The Movable Feast", "restaurant\_id" : "40361606" }

Type "it" for more

4. Write a MongoDB query to display the fields restaurant\_id, name, borough and zip code, but exclude the field \_id for all the documents in the collection restaurant.

=> db.addresses.aggregate({$project:{restaurant\_id:1,name:1,borough:1,cuisine:1,\_id:0,"address.zipcode":1}})

{ "address" : { "zipcode" : "10462" }, "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Morris Park Bake Shop", "restaurant\_id" : "30075445" }

{ "address" : { "zipcode" : "11225" }, "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "Wendy'S", "restaurant\_id" : "30112340" }

{ "address" : { "zipcode" : "10019" }, "borough" : "Manhattan", "cuisine" : "Irish", "name" : "Dj Reynolds Pub And Restaurant", "restaurant\_id" : "30191841" }

{ "address" : { "zipcode" : "11224" }, "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Riviera Caterer", "restaurant\_id" : "40356018" }

{ "address" : { "zipcode" : "11374" }, "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Tov Kosher Kitchen", "restaurant\_id" : "40356068" }

{ "address" : { "zipcode" : "11369" }, "borough" : "Queens", "cuisine" : "American ", "name" : "Brunos On The Boulevard", "restaurant\_id" : "40356151" }

{ "address" : { "zipcode" : "10314" }, "borough" : "Staten Island", "cuisine" : "Jewish/Kosher", "name" : "Kosher Island", "restaurant\_id" : "40356442" }

{ "address" : { "zipcode" : "11234" }, "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Wilken'S Fine Food", "restaurant\_id" : "40356483" }

{ "address" : { "zipcode" : "11219" }, "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Regina Caterers", "restaurant\_id" : "40356649" }

{ "address" : { "zipcode" : "11226" }, "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Taste The Tropics Ice Cream", "restaurant\_id" : "40356731" }

{ "address" : { "zipcode" : "10460" }, "borough" : "Bronx", "cuisine" : "American ", "name" : "Wild Asia", "restaurant\_id" : "40357217" }

{ "address" : { "zipcode" : "11214" }, "borough" : "Brooklyn", "cuisine" : "American ", "name" : "C & C Catering Service", "restaurant\_id" : "40357437" }

{ "address" : { "zipcode" : "11208" }, "borough" : "Brooklyn", "cuisine" : "Chinese", "name" : "May May Kitchen", "restaurant\_id" : "40358429" }

{ "address" : { "zipcode" : "10065" }, "borough" : "Manhattan", "cuisine" : "American ", "name" : "1 East 66Th Street Kitchen", "restaurant\_id" : "40359480" }

{ "address" : { "zipcode" : "11223" }, "borough" : "Brooklyn", "cuisine" : "Jewish/Kosher", "name" : "Seuda Foods", "restaurant\_id" : "40360045" }

{ "address" : { "zipcode" : "11218" }, "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant\_id" : "40360076" }

{ "address" : { "zipcode" : "11004" }, "borough" : "Queens", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant\_id" : "40361322" }

{ "address" : { "zipcode" : "11209" }, "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Nordic Delicacies", "restaurant\_id" : "40361390" }

{ "address" : { "zipcode" : "10021" }, "borough" : "Manhattan", "cuisine" : "American ", "name" : "Glorious Food", "restaurant\_id" : "40361521" }

{ "address" : { "zipcode" : "11215" }, "borough" : "Brooklyn", "cuisine" : "American ", "name" : "The Movable Feast", "restaurant\_id" : "40361606" }

Type "it" for more

5. Write a MongoDB query to display the first 5 restaurant which is in the borough Bronx.

=> db.addresses.aggregate([{$match:{borough:'Bronx'}},{$project:{name:1,borough:1}},{$limit:5}])

{ "\_id" : ObjectId("61f2711fc4002f57a303c2b3"), "borough" : "Bronx", "name" : "Morris Park Bake Shop" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2bd"), "borough" : "Bronx", "name" : "Wild Asia" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2d2"), "borough" : "Bronx", "name" : "Carvel Ice Cream" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2d6"), "borough" : "Bronx", "name" : "Happy Garden" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2e8"), "borough" : "Bronx", "name" : "Happy Garden" }

6. Write a MongoDB query to display all the restaurant which is in the borough Bronx.

=> db.addresses.aggregate([{$match:{borough:'Bronx'}},{$project:{borough:1,name:1}}])

{ "\_id" : ObjectId("61f2711fc4002f57a303c2b3"), "borough" : "Bronx", "name" : "Morris Park Bake Shop" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2bd"), "borough" : "Bronx", "name" : "Wild Asia" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2d2"), "borough" : "Bronx", "name" : "Carvel Ice Cream" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2d6"), "borough" : "Bronx", "name" : "Happy Garden" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2e8"), "borough" : "Bronx", "name" : "Happy Garden" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c2f0"), "borough" : "Bronx", "name" : "Manhem Club" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c308"), "borough" : "Bronx", "name" : "The New Starling Athletic Club Of The Bronx" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c320"), "borough" : "Bronx", "name" : "Yankee Tavern" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c333"), "borough" : "Bronx", "name" : "Mcdwyers Pub" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c34c"), "borough" : "Bronx", "name" : "The Punch Bowl" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c357"), "borough" : "Bronx", "name" : "Munchtime" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c359"), "borough" : "Bronx", "name" : "Ihop" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c35e"), "borough" : "Bronx", "name" : "Lulu'S Coffee Shop" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c364"), "borough" : "Bronx", "name" : "Marina Delray" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c389"), "borough" : "Bronx", "name" : "The Lark'S Nest" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c38a"), "borough" : "Bronx", "name" : "Terrace Cafe" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c38b"), "borough" : "Bronx", "name" : "African Terrace" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c38c"), "borough" : "Bronx", "name" : "Cool Zone" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c38d"), "borough" : "Bronx", "name" : "Beaver Pond" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c38e"), "borough" : "Bronx", "name" : "African Market (Baboon Cafe)" }

Type "it" for more

7. Write a MongoDB query to display the next 5 restaurants after skipping first 5 which are in the borough Bronx.

=> db.addresses.aggregate([{$match:{borough:'Bronx'}},{$skip:5},{$project:{borough:1,name:1}}])

{ "\_id" : ObjectId("61f2711fc4002f57a303c2f0"), "borough" : "Bronx", "name" : "Manhem Club" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c308"), "borough" : "Bronx", "name" : "The New Starling Athletic Club Of The Bronx" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c320"), "borough" : "Bronx", "name" : "Yankee Tavern" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c333"), "borough" : "Bronx", "name" : "Mcdwyers Pub" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c34c"), "borough" : "Bronx", "name" : "The Punch Bowl" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c357"), "borough" : "Bronx", "name" : "Munchtime" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c359"), "borough" : "Bronx", "name" : "Ihop" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c35e"), "borough" : "Bronx", "name" : "Lulu'S Coffee Shop" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c364"), "borough" : "Bronx", "name" : "Marina Delray" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c389"), "borough" : "Bronx", "name" : "The Lark'S Nest" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c38a"), "borough" : "Bronx", "name" : "Terrace Cafe" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c38b"), "borough" : "Bronx", "name" : "African Terrace" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c38c"), "borough" : "Bronx", "name" : "Cool Zone" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c38d"), "borough" : "Bronx", "name" : "Beaver Pond" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c38e"), "borough" : "Bronx", "name" : "African Market (Baboon Cafe)" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c3ae"), "borough" : "Bronx", "name" : "Blue Bay Restaurant" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c3b6"), "borough" : "Bronx", "name" : "Seashore Restaurant" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c3c0"), "borough" : "Bronx", "name" : "Bronx Grill" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c3c3"), "borough" : "Bronx", "name" : "Johnny'S Reef Restaurant" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c3c5"), "borough" : "Bronx", "name" : "White Castle" }

Type "it" for more

8. Write a MongoDB query to find the restaurants who achieved a score more than 90.

=> db.addresses.aggregate([{$match:{"grades.score":{$gt:90}}},{$project:{name:1}}])

{ "\_id" : ObjectId("61f2711fc4002f57a303c411"), "name" : "Murals On 54/Randolphs'S" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c4b2"), "name" : "Gandhi" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c615"), "name" : "Bella Napoli" }

9. Write a MongoDB query to find the restaurants that achieved a score, more than 80 but less than 100.

=> db.addresses.aggregate([{$match:{"grades.score":{$gt:80,$lt:100}}},{$project:{name:1}}])

{ "\_id" : ObjectId("61f2711fc4002f57a303c411"), "name" : "Murals On 54/Randolphs'S" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c4b2"), "name" : "Gandhi" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c615"), "name" : "Bella Napoli" }

{ "\_id" : ObjectId("61f27125c4002f57a303ce81"), "name" : "West 79Th Street Boat Basin Cafe" }

10. Write a MongoDB query to find the restaurants which locate in latitude value less than -95.754168.

=> db.addresses.aggregate([{$match:{"address.coord.0":{$lt:-95.754168}}},{$project:{name:1}}])

{ "\_id" : ObjectId("61f27122c4002f57a303c8fb"), "name" : "Burger King" }

{ "\_id" : ObjectId("61f27124c4002f57a303cc66"), "name" : "Cascarino'S" }

{ "\_id" : ObjectId("61f27125c4002f57a303d10c"), "name" : "Sports Center At Chelsea Piers (Sushi Bar)" }

11. Write a MongoDB query to 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.addresses.aggregate([{$match:{cuisine:{$ne:"American"},"grades.score":{$gt:70},"address.coord.0":{$lt:-65.754168}}},{$project:{name:1,cuisine:1,"address.coord":1}}])

{ "\_id" : ObjectId("61f2711fc4002f57a303c4b2"), "address" : { "coord" : [ -73.9864626, 40.7266739 ] }, "cuisine" : "Indian", "name" : "Gandhi" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c615"), "address" : { "coord" : [ -73.984758, 40.7457939 ] }, "cuisine" : "Pizza/Italian", "name" : "Bella Napoli" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c620"), "address" : { "coord" : [ -73.9243061, 40.8276297 ] }, "cuisine" : "Latin (Cuban, Dominican, Puerto Rican, South & Central American)", "name" : "El Molino Rojo Restaurant" }

{ "\_id" : ObjectId("61f27122c4002f57a303c770"), "address" : { "coord" : [ -73.94610279999999, 40.7137587 ] }, "cuisine" : "Bakery", "name" : "Fortunato Bros Cafe & Bakery" }

{ "\_id" : ObjectId("61f27124c4002f57a303cd9a"), "address" : { "coord" : [ -73.9772294, 40.7527262 ] }, "cuisine" : "Italian", "name" : "Two Boots Grand Central" }

12. Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American' and achieved a score more than 70 and located in the longitude less than -65.754168.

=> db.addresses.aggregate([{$match:{cuisine:{$ne:"American"},"grades.score":{$gt:70},"address.coord.1":{$lt:-65.754168}}},{$project:{name:1,cuisine:1,"address.coord":1}}])

13. Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American ' and achieved a grade point 'A' not belongs to the borough Brooklyn. The document must be displayed according to the cuisine in descending order.

=> db.addresses.aggregate([{$match:{cuisine:{$ne:"American"},"grades.grade":'A',borough:{$ne:'Brooklyn'}}},{$project:{name:1,cuisine:1,borough:1,"address.coord":1,\_id:0}}])

{ "address" : { "coord" : [ -73.856077, 40.848447 ] }, "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Morris Park Bake Shop" }

{ "address" : { "coord" : [ -73.98513559999999, 40.7676919 ] }, "borough" : "Manhattan", "cuisine" : "Irish", "name" : "Dj Reynolds Pub And Restaurant" }

{ "address" : { "coord" : [ -73.8601152, 40.7311739 ] }, "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Tov Kosher Kitchen" }

{ "address" : { "coord" : [ -74.1377286, 40.6119572 ] }, "borough" : "Staten Island", "cuisine" : "Jewish/Kosher", "name" : "Kosher Island" }

{ "address" : { "coord" : [ -73.7032601, 40.7386417 ] }, "borough" : "Queens", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream" }

{ "address" : { "coord" : [ -73.839297, 40.78147 ] }, "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Sal'S Deli" }

{ "address" : { "coord" : [ -73.9925306, 40.7309346 ] }, "borough" : "Manhattan", "cuisine" : "Delicatessen", "name" : "Bully'S Deli" }

{ "address" : { "coord" : [ -73.94024739999999, 40.7623288 ] }, "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Steve Chu'S Deli & Grocery" }

{ "address" : { "coord" : [ -73.976112, 40.786714 ] }, "borough" : "Manhattan", "cuisine" : "Chicken", "name" : "Harriet'S Kitchen" }

{ "address" : { "coord" : [ -73.8642349, 40.75356 ] }, "borough" : "Queens", "cuisine" : "Chinese", "name" : "Ho Mei Restaurant" }

{ "address" : { "coord" : [ -74.0085357, 40.70620539999999 ] }, "borough" : "Manhattan", "cuisine" : "Turkish", "name" : "The Country Cafe" }

{ "address" : { "coord" : [ -73.84856870000002, 40.8903781 ] }, "borough" : "Bronx", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream" }

{ "address" : { "coord" : [ -73.9998042, 40.7251256 ] }, "borough" : "Manhattan", "cuisine" : "Bakery", "name" : "Olive'S" }

{ "address" : { "coord" : [ -73.8893654, 40.81376179999999 ] }, "borough" : "Bronx", "cuisine" : "Chinese", "name" : "Happy Garden" }

{ "address" : { "coord" : [ -73.902463, 40.694924 ] }, "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Tony'S Deli" }

{ "address" : { "coord" : [ -73.97534999999999, 40.7516269 ] }, "borough" : "Manhattan", "cuisine" : "Sandwiches/Salads/Mixed Buffet", "name" : "Lexler Deli" }

{ "address" : { "coord" : [ -74.1459332, 40.6103714 ] }, "borough" : "Staten Island", "cuisine" : "Delicatessen", "name" : "Bagels N Buns" }

{ "address" : { "coord" : [ -73.8740217, 40.7135015 ] }, "borough" : "Queens", "cuisine" : "Bagels/Pretzels", "name" : "Hot Bagels" }

{ "address" : { "coord" : [ -73.95685019999999, 40.7753401 ] }, "borough" : "Manhattan", "cuisine" : "Continental", "name" : "Lorenzo & Maria'S" }

{ "address" : { "coord" : [ -73.9791458, 40.744328 ] }, "borough" : "Manhattan", "cuisine" : "Pizza", "name" : "Domino'S Pizza" }

Type "it" for more

14. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Wil' as first three letters for its name.

=> db.addresses.aggregate([{$match:{name:{$regex:"^Wil"}}},{$project:{name:1,borough:1,cuisine:1,restaurant\_id:1,\_id:0}}])

{ "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Wilken'S Fine Food", "restaurant\_id" : "40356483" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Wild Asia", "restaurant\_id" : "40357217" }

{ "borough" : "Bronx", "cuisine" : "Pizza", "name" : "Wilbel Pizza", "restaurant\_id" : "40871979" }

15. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'ces' as last three letters for its name.

=> db.addresses.aggregate([{$match:{name:{$regex:"ces$"}}},{$project:{name:1,borough:1,cuisine:1,restaurant\_id:1,\_id:0}}])

{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Pieces", "restaurant\_id" : "40399910" }

{ "borough" : "Queens", "cuisine" : "American ", "name" : "S.M.R Restaurant Services", "restaurant\_id" : "40403857" }

{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Good Shepherd Services", "restaurant\_id" : "40403989" }

{ "borough" : "Queens", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "The Ice Box-Ralph'S Famous Italian Ices", "restaurant\_id" : "40690899" }

{ "borough" : "Brooklyn", "cuisine" : "Jewish/Kosher", "name" : "Alices", "restaurant\_id" : "40782042" }

{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Re: Sources", "restaurant\_id" : "40876068" }

16. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Reg' as three letters somewhere in its name.

=> db.addresses.aggregate([{$match:{name:{$regex:"Wil"}}},{$project:{name:1,borough:1,cuisine:1,restaurant\_id:1,\_id:0}}])

{ "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Wilken'S Fine Food", "restaurant\_id" : "40356483" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Wild Asia", "restaurant\_id" : "40357217" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "The Williamsbridge Tavern", "restaurant\_id" : "40852754" }

{ "borough" : "Bronx", "cuisine" : "Pizza", "name" : "Wilbel Pizza", "restaurant\_id" : "40871979" }

17. Write a MongoDB query to find the restaurants which belong to the borough Bronx and prepared either American or Chinese dish.

=> db.addresses.aggregate([{$match:{borough:'Bronx', $or:[{cuisine:'American '},{cuisine:'Chinese'}]}},{$project:{name:1,restaurant\_id:1,cuisine:1,borough:1,\_id:0}}])

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Wild Asia", "restaurant\_id" : "40357217" }

{ "borough" : "Bronx", "cuisine" : "Chinese", "name" : "Happy Garden", "restaurant\_id" : "40363289" }

{ "borough" : "Bronx", "cuisine" : "Chinese", "name" : "Happy Garden", "restaurant\_id" : "40364296" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Manhem Club", "restaurant\_id" : "40364363" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "The New Starling Athletic Club Of The Bronx", "restaurant\_id" : "40364956" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Yankee Tavern", "restaurant\_id" : "40365499" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "The Punch Bowl", "restaurant\_id" : "40366497" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Munchtime", "restaurant\_id" : "40366748" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Marina Delray", "restaurant\_id" : "40367161" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Cool Zone", "restaurant\_id" : "40368022" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Beaver Pond", "restaurant\_id" : "40368025" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "African Market (Baboon Cafe)", "restaurant\_id" : "40368026" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Blue Bay Restaurant", "restaurant\_id" : "40369087" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Bronx Grill", "restaurant\_id" : "40369601" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "P & K'S Grill", "restaurant\_id" : "40369895" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "John Mulligan'S Fireside Pub", "restaurant\_id" : "40370150" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Quality Cafe & Restaurant", "restaurant\_id" : "40374259" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Riverdale Diner", "restaurant\_id" : "40380253" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Castlehill Diner", "restaurant\_id" : "40382517" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Short Stop Restaurant", "restaurant\_id" : "40383819" }

Type "it" for more

18. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which belong to the borough Staten Island or Queens or Bronxor Brooklyn.

=> db.addresses.aggregate([{$match:{borough:{$in:['Queens','Staten Island','Bronx','Brooklyn']}}},{$project:{name:1,restaurant\_id:1,cuisine:1,borough:1,\_id:0}}])

{ "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Morris Park Bake Shop", "restaurant\_id" : "30075445" }

{ "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "Wendy'S", "restaurant\_id" : "30112340" }

{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Riviera Caterer", "restaurant\_id" : "40356018" }

{ "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Tov Kosher Kitchen", "restaurant\_id" : "40356068" }

{ "borough" : "Queens", "cuisine" : "American ", "name" : "Brunos On The Boulevard", "restaurant\_id" : "40356151" }

{ "borough" : "Staten Island", "cuisine" : "Jewish/Kosher", "name" : "Kosher Island", "restaurant\_id" : "40356442" }

{ "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Wilken'S Fine Food", "restaurant\_id" : "40356483" }

{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Regina Caterers", "restaurant\_id" : "40356649" }

{ "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Taste The Tropics Ice Cream", "restaurant\_id" : "40356731" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Wild Asia", "restaurant\_id" : "40357217" }

{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "C & C Catering Service", "restaurant\_id" : "40357437" }

{ "borough" : "Brooklyn", "cuisine" : "Chinese", "name" : "May May Kitchen", "restaurant\_id" : "40358429" }

{ "borough" : "Brooklyn", "cuisine" : "Jewish/Kosher", "name" : "Seuda Foods", "restaurant\_id" : "40360045" }

{ "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant\_id" : "40360076" }

{ "borough" : "Queens", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant\_id" : "40361322" }

{ "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Nordic Delicacies", "restaurant\_id" : "40361390" }

{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "The Movable Feast", "restaurant\_id" : "40361606" }

{ "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Sal'S Deli", "restaurant\_id" : "40361618" }

{ "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Steve Chu'S Deli & Grocery", "restaurant\_id" : "40361998" }

{ "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "White Castle", "restaurant\_id" : "40362344" }

Type "it" for more

19. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which are not belonging to the borough Staten Island or Queens or Bronxor Brooklyn.

=> db.addresses.aggregate([{$match:{borough:{$nin:['Queens','Staten Island','Bronx','Brooklyn']}}},{$project:{name:1,restaurant\_id:1,cuisine:1,borough:1,\_id:0}}])

{ "borough" : "Manhattan", "cuisine" : "Irish", "name" : "Dj Reynolds Pub And Restaurant", "restaurant\_id" : "30191841" }

{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "1 East 66Th Street Kitchen", "restaurant\_id" : "40359480" }

{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Glorious Food", "restaurant\_id" : "40361521" }

{ "borough" : "Manhattan", "cuisine" : "Delicatessen", "name" : "Bully'S Deli", "restaurant\_id" : "40361708" }

{ "borough" : "Manhattan", "cuisine" : "Chicken", "name" : "Harriet'S Kitchen", "restaurant\_id" : "40362098" }

{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "P & S Deli Grocery", "restaurant\_id" : "40362264" }

{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Angelika Film Center", "restaurant\_id" : "40362274" }

{ "borough" : "Manhattan", "cuisine" : "Turkish", "name" : "The Country Cafe", "restaurant\_id" : "40362715" }

{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Downtown Deli", "restaurant\_id" : "40363021" }

{ "borough" : "Manhattan", "cuisine" : "Bakery", "name" : "Olive'S", "restaurant\_id" : "40363151" }

{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Cafe Metro", "restaurant\_id" : "40363298" }

{ "borough" : "Manhattan", "cuisine" : "Sandwiches/Salads/Mixed Buffet", "name" : "Lexler Deli", "restaurant\_id" : "40363426" }

{ "borough" : "Manhattan", "cuisine" : "Continental", "name" : "Lorenzo & Maria'S", "restaurant\_id" : "40363630" }

{ "borough" : "Manhattan", "cuisine" : "Pizza", "name" : "Domino'S Pizza", "restaurant\_id" : "40363644" }

{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Berkely", "restaurant\_id" : "40363685" }

{ "borough" : "Manhattan", "cuisine" : "Pizza", "name" : "Domino'S Pizza", "restaurant\_id" : "40363945" }

{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Spoon Bread Catering", "restaurant\_id" : "40364179" }

{ "borough" : "Manhattan", "cuisine" : "Chicken", "name" : "Texas Rotisserie", "restaurant\_id" : "40364304" }

{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Metropolitan Club", "restaurant\_id" : "40364347" }

{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "Palm Restaurant", "restaurant\_id" : "40364355" }

Type "it" for more

20. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which achieved a score which is not more than 10.

=> db.addresses.aggregate([{$match:{"grades.score":{$not:{$lt:10}}}},{$project:{name:1,restaurant\_id:1,cuisine:1,borough:1,\_id:0}}])

{ "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Tov Kosher Kitchen", "restaurant\_id" : "40356068" }

{ "borough" : "Queens", "cuisine" : "Bagels/Pretzels", "name" : "Hot Bagels", "restaurant\_id" : "40363565" }

{ "borough" : "Bronx", "cuisine" : "Chinese", "name" : "Happy Garden", "restaurant\_id" : "40364296" }

{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "21 Club", "restaurant\_id" : "40364362" }

{ "borough" : "Manhattan", "cuisine" : "American ", "name" : "7B Bar", "restaurant\_id" : "40364518" }

{ "borough" : "Manhattan", "cuisine" : "Italian", "name" : "Marchis Restaurant", "restaurant\_id" : "40364668" }

{ "borough" : "Brooklyn", "cuisine" : "American ", "name" : "Melody Lanes", "restaurant\_id" : "40364889" }

{ "borough" : "Manhattan", "cuisine" : "French", "name" : "Le Perigord", "restaurant\_id" : "40365414" }

{ "borough" : "Manhattan", "cuisine" : "Irish", "name" : "Moran'S Chelsea", "restaurant\_id" : "40365526" }

{ "borough" : "Manhattan", "cuisine" : "Italian", "name" : "V & T Restaurant", "restaurant\_id" : "40365577" }

{ "borough" : "Staten Island", "cuisine" : "American ", "name" : "Joyce'S Tavern", "restaurant\_id" : "40365692" }

{ "borough" : "Manhattan", "cuisine" : "Italian", "name" : "Il Vagabondo Restaurant", "restaurant\_id" : "40365709" }

{ "borough" : "Queens", "cuisine" : "Italian", "name" : "Parkside Restaurant", "restaurant\_id" : "40365841" }

{ "borough" : "Bronx", "cuisine" : "Irish", "name" : "Mcdwyers Pub", "restaurant\_id" : "40365893" }

{ "borough" : "Manhattan", "cuisine" : "Chinese", "name" : "Hop Kee Restaurant", "restaurant\_id" : "40365942" }

{ "borough" : "Queens", "cuisine" : "Pizza", "name" : "Margherita Pizza", "restaurant\_id" : "40366002" }

{ "borough" : "Manhattan", "cuisine" : "Italian", "name" : "Nanni Restaurant", "restaurant\_id" : "40366157" }

{ "borough" : "Queens", "cuisine" : "Italian", "name" : "Don Peppe", "restaurant\_id" : "40366230" }

{ "borough" : "Queens", "cuisine" : "Irish", "name" : "Irish Cottage", "restaurant\_id" : "40366549" }

{ "borough" : "Staten Island", "cuisine" : "American ", "name" : "Li Greci'S Staaten Restaurant", "restaurant\_id" : "40366711" }

Type "it" for more

21. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which prepared dish except 'American' and 'Chinees' or restaurant's name begins with letter 'Wil'.

=> db.addresses.aggregate([{$match:{$or:[{cuisine:{$nin:['American','Chinese']}},{name:{$regex:"^Wil"}}]}},{$project:{restaurant\_id:1,cuisine:1,borough:1,name:1,\_id:0}}])

{ "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Morris Park Bake Shop", "restaurant\_id" : "30075445" }

{ "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "Wendy'S", "restaurant\_id" : "30112340" }

{ "borough" : "Manhattan", "cuisine" : "Irish", "name" : "Dj Reynolds Pub And Restaurant", "restaurant\_id" : "30191841" }

{ "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Tov Kosher Kitchen", "restaurant\_id" : "40356068" }

{ "borough" : "Staten Island", "cuisine" : "Jewish/Kosher", "name" : "Kosher Island", "restaurant\_id" : "40356442" }

{ "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Wilken'S Fine Food", "restaurant\_id" : "40356483" }

{ "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Taste The Tropics Ice Cream", "restaurant\_id" : "40356731" }

{ "borough" : "Bronx", "cuisine" : "American ", "name" : "Wild Asia", "restaurant\_id" : "40357217" }

{ "borough" : "Brooklyn", "cuisine" : "Jewish/Kosher", "name" : "Seuda Foods", "restaurant\_id" : "40360045" }

{ "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant\_id" : "40360076" }

{ "borough" : "Queens", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant\_id" : "40361322" }

{ "borough" : "Brooklyn", "cuisine" : "Delicatessen", "name" : "Nordic Delicacies", "restaurant\_id" : "40361390" }

{ "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Sal'S Deli", "restaurant\_id" : "40361618" }

{ "borough" : "Manhattan", "cuisine" : "Delicatessen", "name" : "Bully'S Deli", "restaurant\_id" : "40361708" }

{ "borough" : "Queens", "cuisine" : "Delicatessen", "name" : "Steve Chu'S Deli & Grocery", "restaurant\_id" : "40361998" }

{ "borough" : "Manhattan", "cuisine" : "Chicken", "name" : "Harriet'S Kitchen", "restaurant\_id" : "40362098" }

{ "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "White Castle", "restaurant\_id" : "40362344" }

{ "borough" : "Manhattan", "cuisine" : "Turkish", "name" : "The Country Cafe", "restaurant\_id" : "40362715" }

{ "borough" : "Brooklyn", "cuisine" : "Caribbean", "name" : "Shashemene Int'L Restaura", "restaurant\_id" : "40362869" }

{ "borough" : "Bronx", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "name" : "Carvel Ice Cream", "restaurant\_id" : "40363093" }

Type "it" for more

22. Write a MongoDB query to find the restaurant Id, name, and grades for those restaurants which achieved a grade of "A" and scored 11 on an ISODate "2014-08-11T00:00:00Z" among many of survey dates..

=> db.addresses.find({"grades.date": ISODate("2014-08-11T00:00:00Z"), "grades.grade":"A" , "grades.score" : 11}, {"restaurant\_id" : 1,"name":1,"grades":1})

{ "\_id" : ObjectId("61f2711fc4002f57a303c331"), "grades" : [ { "date" : ISODate("2014-08-11T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2013-07-22T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2013-03-14T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2012-07-02T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2012-02-02T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2011-08-24T00:00:00Z"), "grade" : "A", "score" : 11 } ], "name" : "Neary'S Pub", "restaurant\_id" : "40365871" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c40c"), "grades" : [ { "date" : ISODate("2014-08-11T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2013-12-10T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2013-06-10T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2012-06-08T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2012-01-25T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2011-09-13T00:00:00Z"), "grade" : "A", "score" : 12 } ], "name" : "Don Filippo Restaurant", "restaurant\_id" : "40372417" }

{ "\_id" : ObjectId("61f27122c4002f57a303c6d2"), "grades" : [ { "date" : ISODate("2014-08-11T00:00:00Z"), "grade" : "A", "score" : 6 }, { "date" : ISODate("2013-08-29T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2013-04-25T00:00:00Z"), "grade" : "C", "score" : 2 }, { "date" : ISODate("2012-10-23T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2012-04-16T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2011-11-15T00:00:00Z"), "grade" : "A", "score" : 12 } ], "name" : "Mustang Sally'S Restaurant", "restaurant\_id" : "40397374" }

{ "\_id" : ObjectId("61f27122c4002f57a303c8de"), "grades" : [ { "date" : ISODate("2015-01-12T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2014-08-11T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2014-01-14T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2013-02-07T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2012-04-30T00:00:00Z"), "grade" : "A", "score" : 11 } ], "name" : "Club Macanudo (Cigar Bar)", "restaurant\_id" : "40526406" }

{ "\_id" : ObjectId("61f27122c4002f57a303c9d3"), "grades" : [ { "date" : ISODate("2014-08-11T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2014-01-09T00:00:00Z"), "grade" : "B", "score" : 27 }, { "date" : ISODate("2013-06-11T00:00:00Z"), "grade" : "A", "score" : 4 }, { "date" : ISODate("2012-10-09T00:00:00Z"), "grade" : "B", "score" : 19 }, { "date" : ISODate("2012-05-10T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2011-11-23T00:00:00Z"), "grade" : "A", "score" : 9 } ], "name" : "Marino'S Pizza & Restaurant", "restaurant\_id" : "40560917" }

{ "\_id" : ObjectId("61f27124c4002f57a303cb7b"), "grades" : [ { "date" : ISODate("2015-01-05T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2014-08-11T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2014-03-04T00:00:00Z"), "grade" : "B", "score" : 22 }, { "date" : ISODate("2013-09-13T00:00:00Z"), "grade" : "A", "score" : 5 }, { "date" : ISODate("2013-01-16T00:00:00Z"), "grade" : "A", "score" : 11 } ], "name" : "Gene'S Coffee Shop", "restaurant\_id" : "40614916" }

{ "\_id" : ObjectId("61f27124c4002f57a303cce9"), "grades" : [ { "date" : ISODate("2014-08-11T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2014-02-20T00:00:00Z"), "grade" : "A", "score" : 7 }, { "date" : ISODate("2013-08-29T00:00:00Z"), "grade" : "A", "score" : 7 }, { "date" : ISODate("2013-03-29T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2012-03-07T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2011-11-18T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2011-10-12T00:00:00Z"), "grade" : "P", "score" : 4 } ], "name" : "Union Cafe Restaurant", "restaurant\_id" : "40698823" }

{ "\_id" : ObjectId("61f27125c4002f57a303d156"), "grades" : [ { "date" : ISODate("2014-08-11T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2013-08-28T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2012-09-05T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2011-08-25T00:00:00Z"), "grade" : "A", "score" : 7 } ], "name" : "Jojo'S Pizza", "restaurant\_id" : "40892913" }

23. Write a MongoDB query to find the restaurant Id, name and grades for those restaurants where the 2nd element of grades array contains a grade of "A" and score 9 on an ISODate "2014-08-11T00:00:00Z"

=> db.addresses.find({ "grades.1.date": ISODate("2014-08-11T00:00:00Z"), "grades.1.grade":"A" , "grades.1.score" : 9}, {"restaurant\_id" : 1,"name":1,"grades":1}).pretty();

{

"\_id" : ObjectId("61f27122c4002f57a303c8de"),

"grades" : [

{

"date" : ISODate("2015-01-12T00:00:00Z"),

"grade" : "A",

"score" : 10

},

{

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

"grade" : "A",

"score" : 9

},

{

"date" : ISODate("2014-01-14T00:00:00Z"),

"grade" : "A",

"score" : 13

},

{

"date" : ISODate("2013-02-07T00:00:00Z"),

"grade" : "A",

"score" : 10

},

{

"date" : ISODate("2012-04-30T00:00:00Z"),

"grade" : "A",

"score" : 11

}

],

"name" : "Club Macanudo (Cigar Bar)",

"restaurant\_id" : "40526406"

}

24. Write a MongoDB query to find the restaurant Id, name, address and geographical location for those restaurants where 2nd element of coord array contains a value which is more than 42 and upto 52..

=> db.addresses.find({"address.coord.1": {$gt : 42, $lte : 52}},{"restaurant\_id" : 1,"name":1,"address":1,"coord":1});

{ "\_id" : ObjectId("61f2711fc4002f57a303c581"), "address" : { "building" : "1", "coord" : [ -0.7119979, 51.6514664 ], "street" : "Pennplaza E, Penn Sta", "zipcode" : "10001" }, "name" : "T.G.I. Fridays", "restaurant\_id" : "40388936" }

{ "\_id" : ObjectId("61f2711fc4002f57a303c555"), "address" : { "building" : "47", "coord" : [ -78.877224, 42.89546199999999 ], "street" : "Broadway @ Trinity Pl", "zipcode" : "10006" }, "name" : "T.G.I. Friday'S", "restaurant\_id" : "40387990" }

{ "\_id" : ObjectId("61f27122c4002f57a303c7da"), "address" : { "building" : "3000", "coord" : [ -87.86567699999999, 42.61150920000001 ], "street" : "47 Avenue", "zipcode" : "11101" }, "name" : "Di Luvio'S Deli", "restaurant\_id" : "40402284" }

{ "\_id" : ObjectId("61f27122c4002f57a303ca0f"), "address" : { "building" : "21972199", "coord" : [ -78.589606, 42.8912372 ], "street" : "Broadway", "zipcode" : "10024" }, "name" : "La Caridad 78", "restaurant\_id" : "40568285" }

{ "\_id" : ObjectId("61f27125c4002f57a303d0e1"), "address" : { "building" : "7981", "coord" : [ -84.9751215, 45.4713351 ], "street" : "Hoyt Street", "zipcode" : "11201" }, "name" : "Bijan'S", "restaurant\_id" : "40876618" }

{ "\_id" : ObjectId("61f27125c4002f57a303d0f6"), "address" : { "building" : "0", "coord" : [ -88.0778799, 42.4154769 ], "street" : "& Grand Central", "zipcode" : "10017" }, "name" : "Hyatt, Ny Central/Room Service", "restaurant\_id" : "40879243" }

{ "\_id" : ObjectId("61f27125c4002f57a303d10c"), "address" : { "building" : "60", "coord" : [ -111.9975205, 42.0970258 ], "street" : "West Side Highway", "zipcode" : "10006" }, "name" : "Sports Center At Chelsea Piers (Sushi Bar)", "restaurant\_id" : "40882356" }

25. Write a MongoDB query to arrange the name of the restaurants in ascending order along with all the columns.

=> db.addresses.find({}).sort({"name":1})

26. Write a MongoDB query to arrange the name of the restaurants in descending along with all the columns.

=> db.addresses.find({}).sort({"name":-1})

27. Write a MongoDB query to arranged the name of the cuisine in ascending order and for that same cuisine borough should be in descending order.

=> db.addresses.find({}).sort({"cuisine":1,"borough":-1})

28. Write a MongoDB query to know whether all the addresses contains the street or not.

=> db.addresses.find({"address.street":{$exists:true}})

29. Write a MongoDB query which will select all documents in the restaurants collection where the coord field value is Double.

=> db.addresses.find({"address.coord":{$type:1}})

30. Write a MongoDB query which will select the restaurant Id, name and grades for those restaurants which returns 0 as a remainder after dividing the score by 7.

=> db.addresses.find( {"grades.score" : {$mod : [7,0]} },{"restaurant\_id" : 1,"name":1,"grades":1});

31. Write a MongoDB query to find the restaurant name, borough, longitude and attitude and cuisine for those restaurants which contains 'mon' as three letters somewhere in its name.

=> db.addresses.find({"name":/.\*mon.\*/},{"restaurant\_id" : 1,"name":1,"address.coord":1})

32. Write a MongoDB query to find the restaurant name, borough, longitude and latitude and cuisine for those restaurants which contain 'Mad' as first three letters of its name.

=> db.addresses.find({"name":/^Mad/},{"restaurant\_id" : 1,"name":1,"address.coord":1})