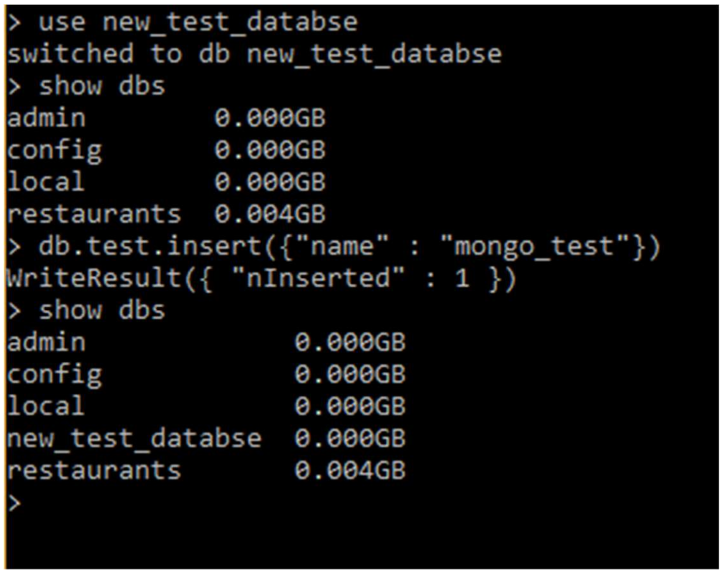
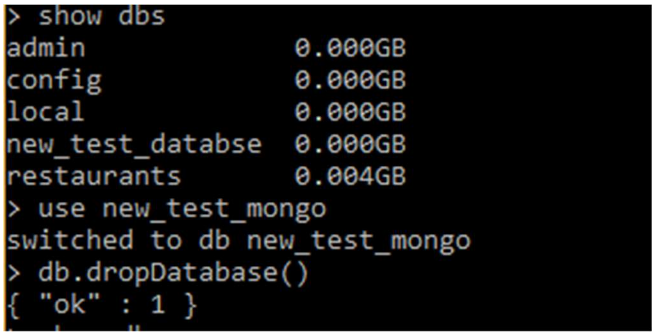
Section 1

1. Create a database



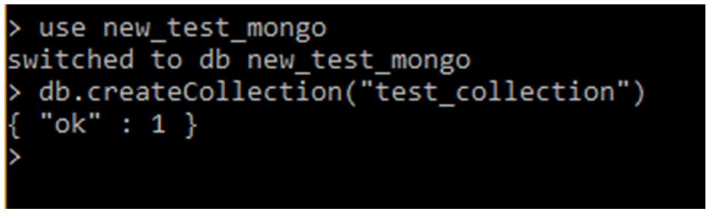
```
> use new_test_databse
switched to db new_test_databse
> show dbs
admin        0.000GB
config       0.000GB
local        0.000GB
restaurants  0.004GB
> db.test.insert({"name" : "mongo_test"})
WriteResult({ "nInserted" : 1 })
> show dbs
admin             0.000GB
config            0.000GB
local             0.000GB
new_test_databse  0.000GB
restaurants       0.004GB
>
```

2. Drop a database



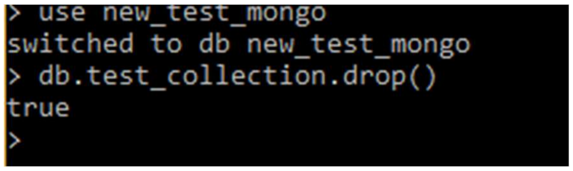
```
> show dbs
admin             0.000GB
config            0.000GB
local             0.000GB
new_test_databse  0.000GB
restaurants       0.004GB
> use new_test_mongo
switched to db new_test_mongo
> db.dropDatabase()
{ "ok" : 1 }
```

3. Creating a collection



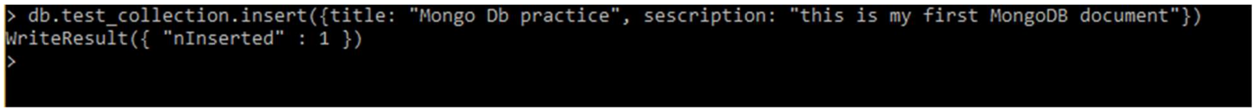
```
> use new_test_mongo
switched to db new_test_mongo
> db.createCollection("test_collection")
{ "ok" : 1 }
>
```

4. Dropping a collection



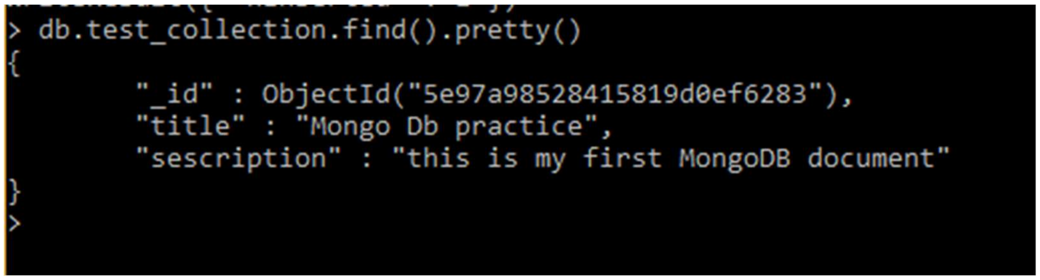
```
> use new_test_mongo
switched to db new_test_mongo
> db.test_collection.drop()
true
>
```

5. Insert a document



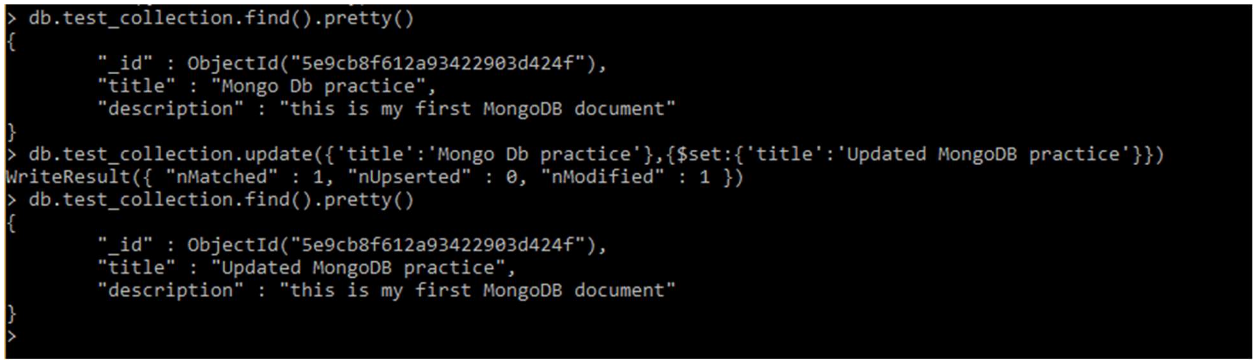
```
> db.test_collection.insert({title: "Mongo Db practice", sescription: "this is my first MongoDB document"})
WriteResult({ "nInserted" : 1 })
>
```

6. Query a document



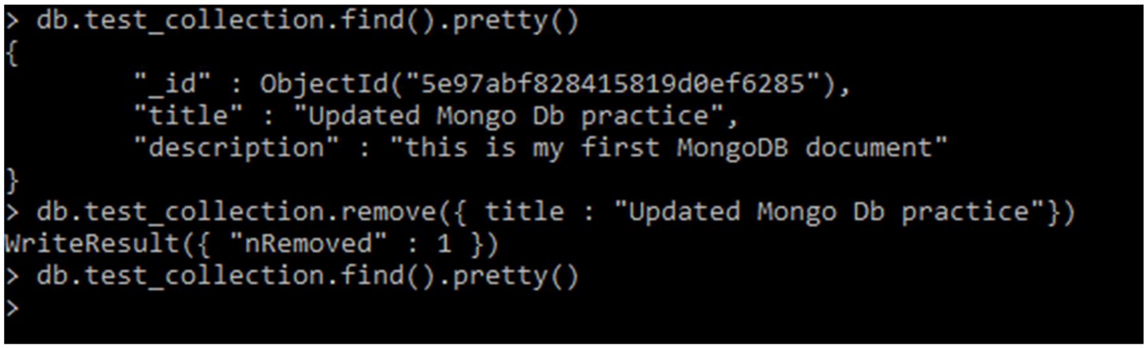
```
> db.test_collection.find().pretty()
{
        "_id" : ObjectId("5e97a98528415819d0ef6283"),
        "title" : "Mongo Db practice",
        "sescription" : "this is my first MongoDB document"
}
>
```

7. Update a document



```
> db.test_collection.find().pretty()
{
        "_id" : ObjectId("5e9cb8f612a93422903d424f"),
        "title" : "Mongo Db practice",
        "description" : "this is my first MongoDB document"
}
> db.test_collection.update({'title':'Mongo Db practice'},{$set:{'title':'Updated MongoDB practice'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.test_collection.find().pretty()
{
        "_id" : ObjectId("5e9cb8f612a93422903d424f"),
        "title" : "Updated MongoDB practice",
        "description" : "this is my first MongoDB document"
}
>
```

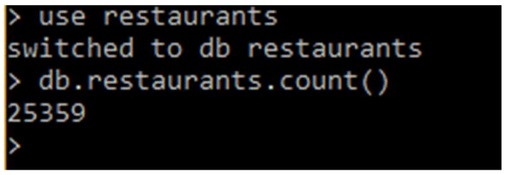
8. Delete a document



```
> db.test_collection.find().pretty()
{
        "_id" : ObjectId("5e97abf828415819d0ef6285"),
        "title" : "Updated Mongo Db practice",
        "description" : "this is my first MongoDB document"
}
> db.test_collection.remove({ title : "Updated Mongo Db practice"})
WriteResult({ "nRemoved" : 1 })
> db.test_collection.find().pretty()
>
```

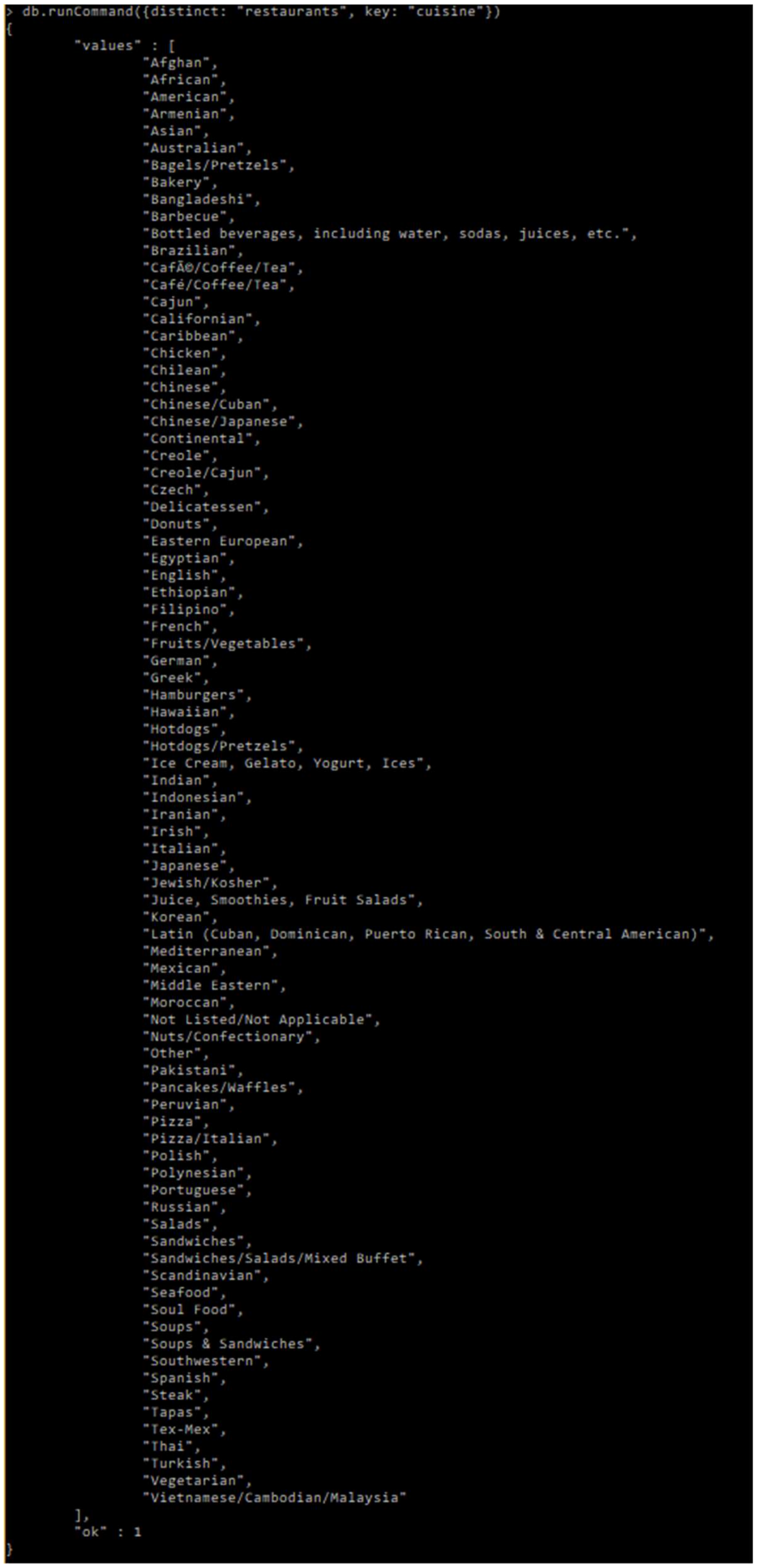
Section 2

Question 1:



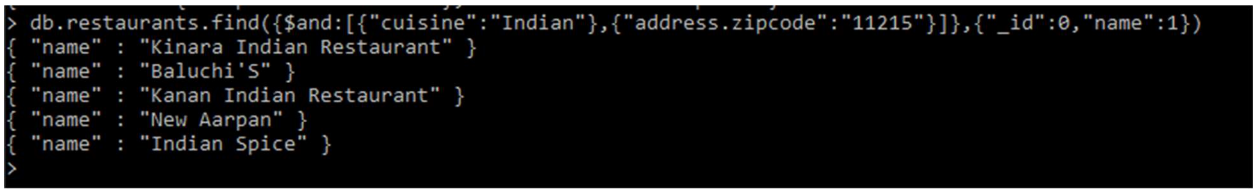
```
> use restaurants
switched to db restaurants
> db.restaurants.count()
25359
>
```

Question 2:



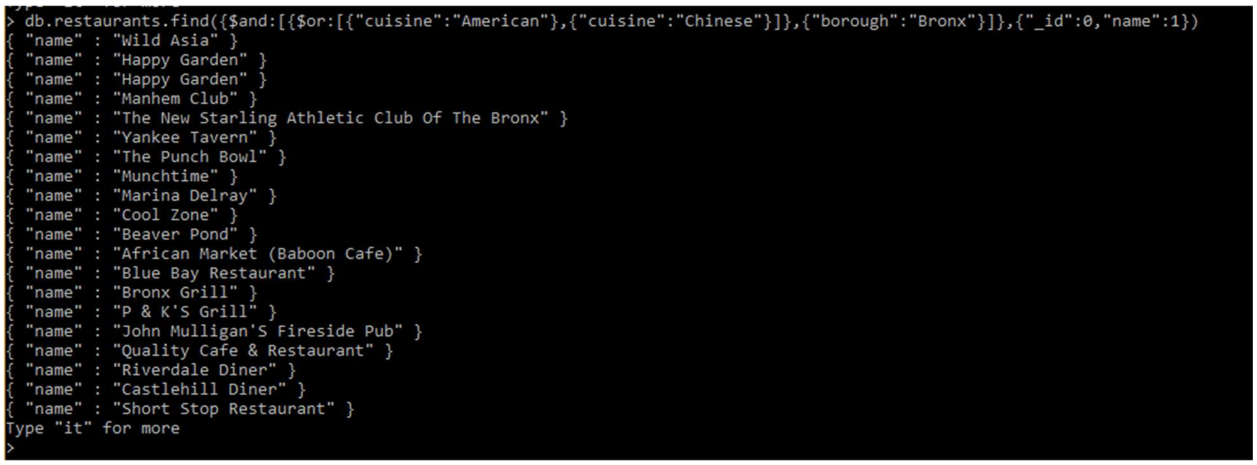
```
> db.runCommand({distinct: "restaurants", key: "cuisine"})
{
        "values" : [
                "Afghan",
                "African",
                "American",
                "Armenian",
                "Asian",
                "Australian",
                "Bagels/Pretzels",
                "Bakery",
                "Bangladeshi",
                "Barbecue",
                "Bottled beverages, including water, sodas, juices, etc.",
                "Brazilian",
                "CafÃ©/Coffee/Tea",
                "Café/Coffee/Tea",
                "Cajun",
                "Californian",
                "Caribbean",
                "Chicken",
                "Chilean",
                "Chinese",
                "Chinese/Cuban",
                "Chinese/Japanese",
                "Continental",
                "Creole",
                "Creole/Cajun",
                "Czech",
                "Delicatessen",
                "Donuts",
                "Eastern European",
                "Egyptian",
                "English",
                "Ethiopian",
                "Filipino",
                "French",
                "Fruits/Vegetables",
                "German",
                "Greek",
                "Hamburgers",
                "Hawaiian",
                "Hotdogs",
                "Hotdogs/Pretzels",
                "Ice Cream, Gelato, Yogurt, Ices",
                "Indian",
                "Indonesian",
                "Iranian",
                "Irish",
                "Italian",
                "Japanese",
                "Jewish/Kosher",
                "Juice, Smoothies, Fruit Salads",
                "Korean",
                "Latin (Cuban, Dominican, Puerto Rican, South & Central American)",
                "Mediterranean",
                "Mexican",
                "Middle Eastern",
                "Moroccan",
                "Not Listed/Not Applicable",
                "Nuts/Confectionary",
                "Other",
                "Pakistani",
                "Pancakes/Waffles",
                "Peruvian",
                "Pizza",
                "Pizza/Italian",
                "Polish",
                "Polynesian",
                "Portuguese",
                "Russian",
                "Salads",
                "Sandwiches",
                "Sandwiches/Salads/Mixed Buffet",
                "Scandinavian",
                "Seafood",
                "Soul Food",
                "Soups",
                "Soups & Sandwiches",
                "Southwestern",
                "Spanish",
                "Steak",
                "Tapas",
                "Tex-Mex",
                "Thai",
                "Turkish",
                "Vegetarian",
                "Vietnamese/Cambodian/Malaysia"
        ],
        "ok" : 1
}
```

Question 3:



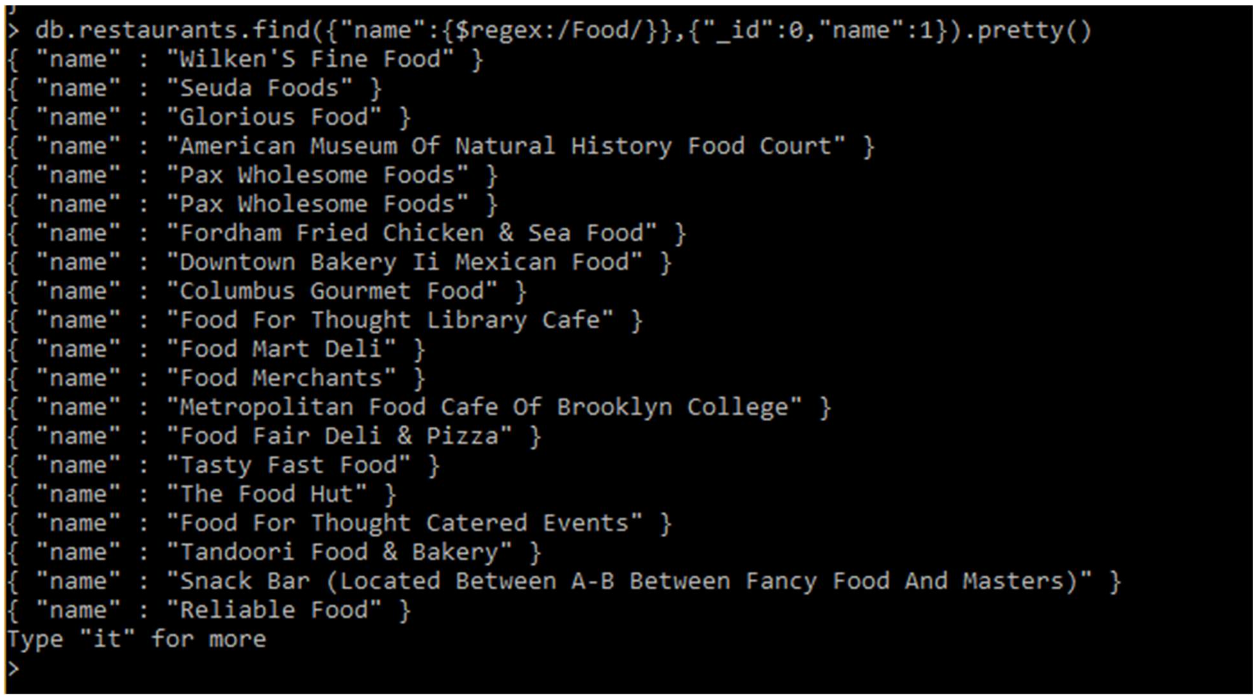
```
> db.restaurants.find({$and:[{"cuisine":"Indian"},{"address.zipcode":"11215"}]},{"_id":0,"name":1})
{ "name" : "Kinara Indian Restaurant" }
{ "name" : "Baluchi'S" }
{ "name" : "Kanan Indian Restaurant" }
{ "name" : "New Aarpan" }
{ "name" : "Indian Spice" }
>
```

Question 4:



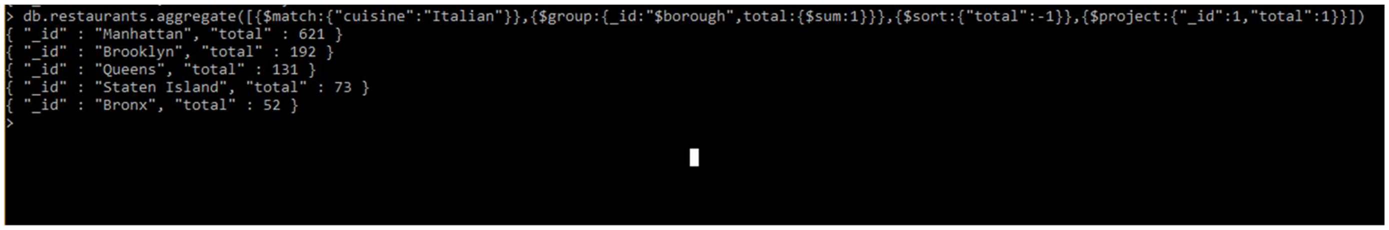
```
> db.restaurants.find({$and:[{$or:[{"cuisine":"American"},{"cuisine":"Chinese"}]},{"borough":"Bronx"}]},{"_id":0,"name":1})
{ "name" : "Wild Asia" }
{ "name" : "Happy Garden" }
{ "name" : "Happy Garden" }
{ "name" : "Manhem Club" }
{ "name" : "The New Starling Athletic Club Of The Bronx" }
{ "name" : "Yankee Tavern" }
{ "name" : "The Punch Bowl" }
{ "name" : "Munchtime" }
{ "name" : "Marina Delray" }
{ "name" : "Cool Zone" }
{ "name" : "Beaver Pond" }
{ "name" : "African Market (Baboon Cafe)" }
{ "name" : "Blue Bay Restaurant" }
{ "name" : "Bronx Grill" }
{ "name" : "P & K'S Grill" }
{ "name" : "John Mulligan'S Fireside Pub" }
{ "name" : "Quality Cafe & Restaurant" }
{ "name" : "Riverdale Diner" }
{ "name" : "Castlehill Diner" }
{ "name" : "Short Stop Restaurant" }
Type "it" for more
>
```

Question 5:



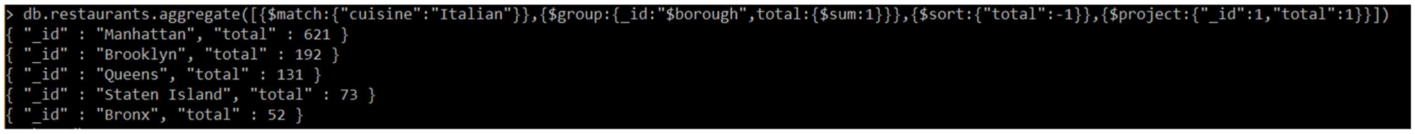
```
> db.restaurants.find({"name":{$regex:/Food/}},{"_id":0,"name":1}).pretty()
{ "name" : "Wilken'S Fine Food" }
{ "name" : "Seuda Foods" }
{ "name" : "Glorious Food" }
{ "name" : "American Museum Of Natural History Food Court" }
{ "name" : "Pax Wholesome Foods" }
{ "name" : "Pax Wholesome Foods" }
{ "name" : "Fordham Fried Chicken & Sea Food" }
{ "name" : "Downtown Bakery Ii Mexican Food" }
{ "name" : "Columbus Gourmet Food" }
{ "name" : "Food For Thought Library Cafe" }
{ "name" : "Food Mart Deli" }
{ "name" : "Food Merchants" }
{ "name" : "Metropolitan Food Cafe Of Brooklyn College" }
{ "name" : "Food Fair Deli & Pizza" }
{ "name" : "Tasty Fast Food" }
{ "name" : "The Food Hut" }
{ "name" : "Food For Thought Catered Events" }
{ "name" : "Tandoori Food & Bakery" }
{ "name" : "Snack Bar (Located Between A-B Between Fancy Food And Masters)" }
{ "name" : "Reliable Food" }
Type "it" for more
>
```

Question 6:



```
> db.restaurants.aggregate([{$match:{"cuisine":"Italian"}},{$group:{_id:"$borough",total:{$sum:1}}},{$sort:{"total":-1}},{$project:{"_id":1,"total":1}}])
{ "_id" : "Manhattan", "total" : 621 }
{ "_id" : "Brooklyn", "total" : 192 }
{ "_id" : "Queens", "total" : 131 }
{ "_id" : "Staten Island", "total" : 73 }
{ "_id" : "Bronx", "total" : 52 }
>
```

Question 7:



```
> db.restaurants.aggregate([{$match:{"cuisine":"Italian"}},{$group:{_id:"$borough",total:{$sum:1}}},{$sort:{"total":-1}},{$project:{"_id":1,"total":1}}])
{ "_id" : "Manhattan", "total" : 621 }
{ "_id" : "Brooklyn", "total" : 192 }
{ "_id" : "Queens", "total" : 131 }
{ "_id" : "Staten Island", "total" : 73 }
{ "_id" : "Bronx", "total" : 52 }
```