

1.

```
>use new_mongo_db
```

```
[> use new_mongo_db  
switched to db new_mongo_db
```

2.

```
>db.dropDatabase()  
> db.dropDatabase()  
{ "ok" : 1 }
```

3.

```
> db.createCollection("test_collection")  
> db.createCollection("mycol", { capped : true, autoIndexId : true, size : 6142800, max : 10000})
```

```
[> use new_mongo_db  
switched to db new_mongo_db  
[> db.createCollection("test_collection")  
{ "ok" : 1 }  
[> db.createCollection("mycol", { capped : true, autoIndexId : true, size : 6142800, max : 10000})  
{  
  "note" : "The autoIndexId option is deprecated and will be removed in a future release",  
  "ok" : 1  
}
```

4.

```
> db.test_collection.drop()  
> db.mycol.drop()
```

```
> db.test_collection.drop()  
true  
> db.mycol.drop()  
true
```

5.

```
> db.test_collection.insert({title: "Mongo Db practice", description: "This is my first MongoDB document"})
```

```
> db.mycollection.insert({title: "Mongo Db practice", description: "This is my first MongoDB document"})
```

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

6.

```
> db.mycollection.find().pretty()
```

```
[> db.mycollection.find().pretty()
{
  "_id" : ObjectId("607ccfa5b8aa2fa011c5cdb2"),
  "title" : "Mongo Db practice",
  "description" : "This is my first MongoDB document"
}
>
```

7.

```
> db.mycollection.update({"title": "Mongo Db practice"}, {$set: {"title": "New MongoDB practice"}})
```

```
[> db.mycollection.update({"title": "Mongo Db practice"}, {$set: {"title": "New MongoDB practice"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

8.

```
> db.mycollection.remove({status: "P"},1)
```

```
> db.mycollection.remove({status: "P"})
```

```
[> db.mycollection.remove({status: "P"},1)
WriteResult({ "nRemoved" : 0 })
[> db.mycollection.remove({status: "P"})
WriteResult({ "nRemoved" : 0 })
```

Section2:

1.

```
> db.restaurant.find().count()
```

```
[> db.restaurant.find().count()
25359
_
```

2.

```
db.restaurant.distinct( "cuisine" )
```

```
> db.restaurant.distinct( "cuisine" )
[
  "Afghan",
  "African",
  "American",
  "Armenian",
  "Asian",
  "Australian",
  "Bagels/Pretzels",
  "Bakery",
  "Bangladeshi",
  "Barbecue",
  "Bottled beverages, including water, sodas, juices, etc.",
  "Brazilian",
  "Café/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",
```

3.

```
>db.restaurant.find({$and:[{"cuisine":"Italian"}, {"address.zipcode":"10305"}]}, {"_id":0, "name":1})
```

```
[> db.restaurant.find({$and:[{"cuisine":"Italian"}, {"address.zipcode":"10305"}]}, {"_id":0, "name":1})
{ "name" : "Crystal Room" }
{ "name" : "Cafe Luna" }
{ "name" : "Cafe Bella Vita" }
{ "name" : "Trattoria Romana" }
{ "name" : "Bocelli" }
{ "name" : "Joe & John Toto'S Restaurant" }
{ "name" : "Ninos Restaurant" }
{ "name" : "South Fin Grill/The Vanderbilt" }
{ "name" : "Da Noi Ii" }
{ "name" : "Italianissimo Restaurant Cafe Bar" }
{ "name" : "Cielo Ristorante" }
{ "name" : "Basilio'S Inn Restaurant" }
{ "name" : "Goodfella'S Pizzeria" }
{ "name" : "Bin No 5" }
{ "name" : "Michael'S Meatballs & Martinis" }
```

4.

```
> db.restaurant.aggregate({"$match" : {"cuisine" : "Greek"}}, {"$group": {"_id" : "$borough",  
"number" : {$sum : 1}}})
```

```
> db.restaurant.aggregate({"$match" : {"cuisine" : "Greek"}}, {"$group": {"_id" : "$borough", "number" : {$sum : 1}}})  
{ "_id" : "Brooklyn", "number" : 14 }  
{ "_id" : "Queens", "number" : 58 }  
{ "_id" : "Bronx", "number" : 4 }  
{ "_id" : "Manhattan", "number" : 35 }
```

5.

```
> db.restaurant.find({"name": {$regex: /Pho/}}, {"name": 1, _id: 0})  
> db.restaurant.find({"name": {$regex: /Pho/}}, {"name": 1, _id: 0})  
{ "name" : "Phoenix Garden" }  
{ "name" : "Pho Bac Vietnamese Seafood Cuisine" }  
{ "name" : "Pho Hoai Bay Ridge" }  
{ "name" : "Pho Bang Restaurant" }  
{ "name" : "Phoebe'S" }  
{ "name" : "Phoebe'S" }  
{ "name" : "Pho 32 & Shabu" }  
{ "name" : "Pho Bang Restaurant" }  
{ "name" : "Pho Viet-Nam Restaurant" }  
{ "name" : "Golden Phoenix" }  
{ "name" : "Pho Bang Restaurant" }  
{ "name" : "Pho Grand" }  
{ "name" : "Pho Vietnamese Restaurant" }  
{ "name" : "Baquette Pho Sure" }  
{ "name" : "Pho Hoai Rest" }  
{ "name" : "International Center Of Photography" }  
{ "name" : "Pho Mac Vietnamese Cuisine" }  
{ "name" : "Phoenix" }  
{ "name" : "Pho Hoang" }  
{ "name" : "Pho Viet" }
```

6.

```
> db.restaurant.aggregate({"$match" : {"cuisine" : "Mexican"}}, {"$project": {"borough" : 1}},  
{"$group": {"_id" : "$borough", "number" : {$sum : 1}}}, {"$sort": {"number": -1}});
```

```
> db.restaurant.aggregate({"$match" : {"cuisine" : "Mexican"}}, {"$project": {"borough" : 1}}, {"$group": {"_id" : "$borough", "number" : {$sum : 1}}}, {"$sort": {"number": -1}});  
{ "_id" : "Manhattan", "number" : 273 }  
{ "_id" : "Brooklyn", "number" : 212 }  
{ "_id" : "Queens", "number" : 144 }  
{ "_id" : "Bronx", "number" : 89 }  
{ "_id" : "Staten Island", "number" : 33 }  
{ "_id" : "Missing", "number" : 1 }
```

7.

```
> db.restaurant.aggregate([{$unwind: "$grades"}, {$match: {cuisine: "Italian", borough:  
"Brooklyn"}}, {$group: {_id: "$_id", name: {$first: "$name"}, totalscore: {$sum: "$grades.score"}},  
{$sort: {totalscore: -1}}, {$limit: 5}])
```

```
> db.restaurant.aggregate([{$unwind: "$grades"}, {$match: {cuisine: "Italian", borough: "Brooklyn"}}, {$group: {_id: "$_id", n
core: -1}}, {$limit: 5}])
{ "_id" : ObjectId("607cd2c8c53c6be1d66c84bd"), "name" : "Joe'S Pizza", "totalscore" : 173 }
{ "_id" : ObjectId("607cd2c8c53c6be1d66c7aca"), "name" : "Anella", "totalscore" : 153 }
{ "_id" : ObjectId("607cd2c8c53c6be1d66c72cb"), "name" : "Tutta Pasta", "totalscore" : 124 }
{ "_id" : ObjectId("607cd2c8c53c6be1d66c627b"), "name" : "Doc Wine Bar", "totalscore" : 103 }
{ "_id" : ObjectId("607cd2c8c53c6be1d66c83a1"), "name" : "Da Nonna Rosa", "totalscore" : 98 }
```