

```

MongoDB shell version v5.0.6
connecting to: mongodb://127.0.0.1:27017/?
compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("71afdf33-75f8-45c9-b1f8-
72e934380377") }
MongoDB server version: 5.0.6
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been
deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
WARNING: some history file lines were truncated at 4095 bytes.
uncaught exception: SyntaxError: unterminated comment :
@(shell):1:0
uncaught exception: SyntaxError: unexpected token: identifier :
@(shell):1:5
uncaught exception: SyntaxError: unexpected token: identifier :
@(shell):1:11
connecting to: mongodb://localhost /27017
Error: couldn't connect to server localhost :27017, connection attempt failed:
HostNotFound: Could not find address for localhost :27017: SocketException: No
such host is known. :
connect@src/mongo/shell/mongo.js:372:17
@(shell):1:6
crud2
counting the number of documents
db.restaurants.find().length();
3772
how many boroughs in the collection?
db.restaurants.distinct('borough').length;
5
name of the 6th (per Mongo ordering by ascending name) Brooklyn restaurant
db.restaurants.find({}, { 'name': 1, '_id' : 0 }).pretty().sort({name:1})[5];
{ "name" : "104-01 Foster Avenue Coffee Shop(Ups)" }
how many restaurants in the collection with the zipcode of 10075
db.restaurants.find({'address.zipcode' : '10075'}).toArray().length;
27
how many restaurants rated as 'A'
db.restaurants.find({grades : {$elemMatch:{'grade':'A'}}}).toArray().length;
3759
list the names, in descending lexicographical order, of restaurants have the
string 'Kosher' anywhere in their name, that DO NOT have 'Chinese' cuisine, have
a score greater than 40
db.restaurants.find({name : /Kosher/, cuisine: {$not: /Chinese/}, grades :
{$elemMatch: {score : {$gt : 40}}}}, {name: 1, _id : 0}).sort({name : -
1}).pretty();
{ "name" : "Kosher Hut Of Brooklyn" }
{ "name" : "Kosher Delight" }
{ "name" : "Amnon Kosher Pizza" }
how many records DONT contain 'street' information
db.restaurants.find({'$or': [{'address.street' : ''}, {'address.street' : {$type :
10}}, {'address.street' : {$exists' : false}}]}).pretty().length();
0
how many restaurants either serve Italian cuisine or are located in the 10075
zip code
db.restaurants.find({'$or' : [{'cuisine': /Italian/}, {'address.zipcode':
'10075'}]}).pretty().length();
448
Pretty print the restaurant named 'Viand Cafe', omitting 'grades' and 'id'
fields
db.restaurants.find({name:'Viand Cafe'}, {'_id' : 0, grades : 0}).pretty();

```

```

{
  "address" : {
    "building" : "1011",
    "coord" : [
      -73.9623333,
      40.7757194
    ],
    "street" : "East 31st Street",
    "zipcode" : "10075"
  },
  "borough" : "Manhattan",
  "cuisine" : "American(New)",
  "name" : "Viand Cafe",
  "restaurant_id" : "40369753",
  "lastModified" : ISODate("2022-03-06T20:59:38.634Z")
}
Change the 'Viand Cafe' record to have the 'cuisine' value set to 'American
(New)' and to have a 'lastModified' field with a value set to the current date
and time
db.restaurants.updateOne({name : 'Viand Cafe'}, { '$set' : {cuisine:
'American(New)'}, '$set' : {lastModified : new Date()}},{upsert : true});
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
change the address of record with restaurant id of 40369753 to "East 31st Street
db.restaurants.updateOne({ 'restaurant_id' : '40369753'}, { '$set' :
{'address.street': 'East 31st Street'}});
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 0 }
pretty print the restaurant named 'Viand Cafe', omitting 'grades' and 'id'
fields
db.restaurants.find({name: 'Viand Cafe'},{'_id' : 0, grades : 0}).pretty();
{
  "address" : {
    "building" : "1011",
    "coord" : [
      -73.9623333,
      40.7757194
    ],
    "street" : "East 31st Street",
    "zipcode" : "10075"
  },
  "borough" : "Manhattan",
  "cuisine" : "American(New)",
  "name" : "Viand Cafe",
  "restaurant_id" : "40369753",
  "lastModified" : ISODate("2022-03-06T21:02:56.055Z")
}
show (only) the zipcode of all restaurants with a cuisine of 'Other'
db.restaurants.find({'cuisine' : 'Other'}, {'address.zipcode' : 1, '_id' :
0}).pretty();
change all restaurants with a cuisine of 'Other' to 'Category TBD'
db.restaurants.updateMany({ 'cuisine' : 'Other'}, { '$set' : {'cuisine':
'Category TBD'}});
{ "acknowledged" : true, "matchedCount" : 0, "modifiedCount" : 0 }
how many restaurants have a cuisine of 'Category TBD'
db.restaurants.find({ 'cuisine' : 'Category TBD'}).length();
4
bye

```