

MongoCRUD3

Asher Kirshtein

“How many restaurants are in the imported collection?”

```
database> db.restaurant.count()  
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.  
25359,  
database>
```

Show three unique cuisine types from the set of restaurants

```
database> db.restaurant.distinct("cuisine").slice(0,3)
[ 'Afghan', 'African', 'American' ]
```

“Return restaurants from the collection.”

```
database> db.restaurant.find({}, { restaurant_id: 1, name: 1, borough: 1, cuisine: 1, _id: 0 }).sort({ restaurant_id: 1 }).limit(3)
[
  {
    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'
  }
]
```

“Return all restaurants whose borough is Brooklyn

```
database> database> database> db.restaurant.find({borough: "Brooklyn", cuisine: "Jewish/Kosher"}, {_id: 1, name: 1, borough: 1, cuisine: 1}).limit(3)
[
  {
    _id: ObjectId("63f42d6bbd42877e78416bb0"),
    borough: 'Brooklyn',
    cuisine: 'American',
    name: 'Regina Caterers'
  },
  {
    _id: ObjectId("63f42d6bbd42877e78416bb1"),
    borough: 'Brooklyn',
    cuisine: 'American',
    name: 'Riviera Caterer'
  },
  {
    _id: ObjectId("63f42d6bbd42877e78416bb2"),
    borough: 'Brooklyn',
    cuisine: 'Ice Cream, Gelato, Yogurt, Ices',
    name: 'Taste The Tropics Ice Cream'
  }
]
database>
```

“Return all restaurants whose borough is Brooklyn and whose cuisine is Jewish/Kosher.”

```
database> database> database> db.restaurant.find({cuisine: "Jewish/Kosher", name:{$regex: "^Piz"} }, {_id: 1, name: 1, borough: 1, cuisine: 1}).limit(3)
database> db.restaurant.find({borough: "Brooklyn", cuisine: "Jewish/Kosher" }, {_id: 1, name: 1, borough: 1, cuisine: 1}).limit(3)

{
  _id: ObjectId("63f42d6bbd42877e78416bc3"),
  borough: 'Brooklyn',
  cuisine: 'Jewish/Kosher',
  name: 'Seuda Foods'
},
{
  _id: ObjectId("63f42d6bbd42877e78416bef"),
  borough: 'Brooklyn',
  cuisine: 'Jewish/Kosher',
  name: 'Kosher Bagel Hole'
},
{
  _id: ObjectId("63f42d6bbd42877e78416c89"),
  borough: 'Brooklyn',
  cuisine: 'Jewish/Kosher',
  name: 'Mill Basin Kosher Deli'
}
database>
```

Return all restaurants whose cuisine is Jewish/Kosher and whose name begins with Piz.

```
database> database> database> db.restaurant.find({borough:{$nin: ["Brooklyn", "Manhattan"]}} , {_id: 1, name: 1, borough: 1, cuisine: 1}).limit(3)
[
  {
    _id: ObjectId("63f42d6bbd42877e7841956b"),
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Pizza Palace Cafe'
  },
  {
    _id: ObjectId("63f42d6bbd42877e7841a379"),
    borough: 'Brooklyn',
    cuisine: 'Jewish/Kosher',
    name: 'Pizza King'
  },
  {
    _id: ObjectId("63f42d6bbd42877e7841a607"),
    borough: 'Brooklyn',
    cuisine: 'Jewish/Kosher',
    name: 'Pizza World Cafe'
  }
]
database>
```

Return all restaurants that are neither in Brooklyn or in Manhattan.

```
database> db.restaurant.find({borough:{$nin: ["Brooklyn", "Manhattan"]}}, {_id: 1, name: 1, borough: 1, cuisine: 1}).limit(3)
[
  {
    _id: ObjectId("63f42d6bbd42877e78416bac"),
    borough: 'Queens',
    cuisine: 'American',
    name: 'Brunos On The Boulevard'
  },
  {
    _id: ObjectId("63f42d6bbd42877e78416bae"),
    borough: 'Queens',
    cuisine: 'Jewish/Kosher',
    name: 'Tov Kosher Kitchen'
  },
  {
    _id: ObjectId("63f42d6bbd42877e78416baf"),
    borough: 'Staten Island',
    cuisine: 'Jewish/Kosher',
    name: 'Kosher Island'
  }
]
database>
```


“Return all restaurants whose cuisine is either Jewish/Kosher or Japanese.

```
database> db.restaurant.find({cuisine: {$in: ["Jewish/Koaher","Japanese"]} }, {_id: 1, name: 1, borough: 1, cuisine: 1}).limit(3)
[
  {
    _id: ObjectId("63f42d6bbd42877e78416ca1"),
    borough: 'Manhattan',
    cuisine: 'Japanese',
    name: 'Hatsuhana Restaurant'
  },
  {
    _id: ObjectId("63f42d6bbd42877e78416ca5"),
    borough: 'Manhattan',
    cuisine: 'Japanese',
    name: 'Nada-Sushi'
  },
  {
    _id: ObjectId("63f42d6bbd42877e78416cb0"),
    borough: 'Manhattan',
    cuisine: 'Japanese',
    name: 'Dan Tempura'
  }
]
database>
```

Return all restaurants whose cuisine is both Jewish/Kosher and American.

```
database> db.restaurant.find({ cuisine: { $all: ["Jewish/Kosher", "American"] } }, {_id: 1, name: 1, borough: 1, cuisine: 1})  
database>
```

“Return all restaurants whose score is greater than 90.”

```
database> db.restaurant.find({"grades.score":{"$gt:90}}, {_id: 1}).limit(3)
[
  { _id: ObjectId("63f42d6bbd42877e78416d0a") },
  { _id: ObjectId("63f42d6bbd42877e78416daa") },
  { _id: ObjectId("63f42d6bbd42877e78416f0d") }
]
database>
```

Return all restaurants whose score is greater than 80 and less than 90.”

```
database> db.restaurant.find({"grades.score":{"$gt:80, $lt:90}}, {_id: 1}).limit(3)
[
  { _id: ObjectId("63f42d6bbd42877e78416d0a") },
  { _id: ObjectId("63f42d6bbd42877e78416daa") },
  { _id: ObjectId("63f42d6bbd42877e78416f0d") }
]
database>
```

Return all restaurants with exactly two grades.”

```
database> db.restaurant.find({grades: {$size: 2}}, {_id: 1}).limit(3)
[
  { _id: ObjectId("63f42d6bbd42877e78416be7") },
  { _id: ObjectId("63f42d6bbd42877e78416cb5") },
  { _id: ObjectId("63f42d6bbd42877e78416cf9") }
]
database>
```

Return all restaurants that ever received a grade of A.

```
]
database> db.restaurant.find({"grades.grade": "A"}, {_id: 1}).limit(3)
[
  { _id: ObjectId("63f42d6bbd42877e78416bac") },
  { _id: ObjectId("63f42d6bbd42877e78416bad") },
  { _id: ObjectId("63f42d6bbd42877e78416bae") }
]
database>
```