Benjamin Katz
MongoCRUD3

1)"How many restaurants are in the imported collection?"

```
test> db.restaurant2.countDocuments()
25359
```

2)"Show three unique cuisine types from the set of restaurants."

```
test> db.restaurant2.aggregate( [ { $group : { _id : "$cuisine" } }, {$limit: 3} ] ).pretty()
[ { _id: 'Indian' }, { _id: 'Soups' }, { _id: 'Chicken' } ]
```

3)"Return restaurants from the collection."

```
test> db.restaurant2.find({},{restaurant_id: 1, name: 1, borough:1, cuisine:1, _id:0
}).sort({restaurant_id:1}).limit(3).pretty()
[
  {
    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'
  }
```

4). "Return all restaurants whose borough is Brooklyn."

```
test> db.restaurant2.find({borough: "Brooklyn"},{restaurant_id: 1, name: 1, borough:1,
cuisine:1, _id:0 }).sort({restaurant_id:1}).limit(3).pretty()
[
  {
    borough: 'Brooklyn',
    cuisine: 'Hamburgers',
    name: "Wendy'S",
    restaurant_id: '30112340'
```

```
  },
  {
    borough: 'Brooklyn',
    cuisine: 'American',
    name: 'Riviera Caterer',
    restaurant_id: '40356018'
  },
  {
    borough: 'Brooklyn',
    cuisine: 'Delicatessen',
    name: "Wilken'S Fine Food",
    restaurant_id: '40356483'
  }
```

5)"Return all restaurants whose borough is Brooklyn and whose cuisine is Jewish/Kosher."

```
test> db.restaurant2.find({borough: "Brooklyn", cuisine: "Jewish/Kosher"},{restaurant_id: 1,
name: 1, borough:1, cuisine:1, _id:0 }).sort({restaurant_id:1}).limit(3).pretty()
[
  {
    borough: 'Brooklyn',
    cuisine: 'Jewish/Kosher',
    name: 'Seuda Foods',
    restaurant_id: '40360045'
  },
  {
    borough: 'Brooklyn',
    cuisine: 'Jewish/Kosher',
    name: 'Kosher Bagel Hole',
    restaurant_id: '40364220'
  },
  {
    borough: 'Brooklyn',
    cuisine: 'Jewish/Kosher',
    name: 'Mill Basin Kosher Deli',
    restaurant_id: '40368207'
  }
```

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

```
test> db.restaurant2.find({name: {$regex:"^Piz.* "}, cuisine: "Jewish/Kosher"},{restaurant_id: 1,
name: 1, borough:1, cuisine:1, _id:0 }).sort({restaurant_id:1}).limit(3).pretty()
[
  {
    borough: 'Queens',
```

```
  cuisine: 'Jewish/Kosher',
  name: 'Pizza Palace Cafe',
  restaurant_id: '41451780'
 },
 {
  borough: 'Brooklyn',
  cuisine: 'Jewish/Kosher',
  name: 'Pizza King',
  restaurant_id: '41611858'
 },
 {
  borough: 'Brooklyn',
  cuisine: 'Jewish/Kosher',
  name: 'Pizza World Cafe',
  restaurant_id: '41630478'
 }
```

7). "Return all restaurants that are neither in Brooklyn or in Manhattan."

```
test> db.restaurant2.find({borough: {$nin: ["Brooklyn", "Manhattan"]}},{restaurant_id: 1, name:
1, borough:1, cuisine:1, _id:0 }).sort({restaurant_id:1}).limit(3).pretty()
[
 {
  borough: 'Bronx',
  cuisine: 'Bakery',
  name: 'Morris Park Bake Shop',
  restaurant_id: '30075445'
 },
 {
  borough: 'Queens',
  cuisine: 'Jewish/Kosher',
  name: 'Tov Kosher Kitchen',
  restaurant_id: '40356068'
 },
 {
  borough: 'Queens',
  cuisine: 'American',
  name: 'Brunos On The Boulevard',
  restaurant_id: '40356151'
 }
```

8)"Return all restaurants whose cuisine is either Jewish/Kosher or Japanese."

```
test> db.restaurant2.find({cuisine: {$in: ["Jewish/Kosher", "Japanese"]}},{restaurant_id: 1,
name: 1, borough:1, cuisine:1, _id:0 }).sort({restaurant_id:1}).limit(3).pretty()
[
```

```
 {
  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: 'Jewish/Kosher',
  name: 'Seuda Foods',
  restaurant_id: '40360045'
 }
```

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

```
test> db.restaurant2.find({cuisine: {$all: ["Jewish/Kosher","American"]}},{restaurant_id: 1, _id:0
}).sort({restaurant_id:1}).limit(3).pretty()
```

10)"Return all restaurants whose score is greater than 90."

```
test> db.restaurant2.find({"grades.score": {$gt: 90}},{restaurant_id: 1, _id:0
}).sort({restaurant_id:1}).limit(3).pretty()
[
 { restaurant_id: '40372466' },
 { restaurant_id: '40381295' },
 { restaurant_id: '40393488' }
```

11). "Return all restaurants whose score is greater than 80 and less than 90."

```
test> db.restaurant2.find({"grades":{$elemMatch:{score:{$gt: 80, $lt:90}}}},{restaurant_id: 1,
_id:0 }).sort({restaurant_id:1}).limit(3).pretty()
[
 { restaurant_id: '40756344' },
 { restaurant_id: '40979431' },
 { restaurant_id: '40987023' }
```

12). "Return all restaurants with exactly two grades."

```
test> db.restaurant2.find({grades: {$size: 2}},{restaurant_id: 1, _id:0
}).sort({restaurant_id:1}).limit(3).pretty()
```

```
[
  { restaurant_id: '40364363' },
  { restaurant_id: '40369608' },
  { restaurant_id: '40371807' }
]
```

13)“Return all restaurants that ever received a grade of A.”

```
test> db.restaurant2.find({"grades.grade": "A"},{restaurant_id: 1, _id:0
}).sort({restaurant_id:1}).limit(3).pretty()
[
  { restaurant_id: '30075445' },
  { restaurant_id: '30112340' },
  { restaurant_id: '30191841' }
]
```