

## Question1.

쿼리:

```
db.primer.find().limit(4)
```

해당 쿼리를 통해, primer collection으로부터 4개의 documents를 출력할 수 있다. 원래 primer.find() 명령어를 실행한다면 전체 쿼리가 출력되지만, limit를 4개로 줌으로서, 4개만 출력된다.

실행화면

```
> db.primer.find().limit(4)
{ "_id" : ObjectId("5bcaca3073c219978a694833"), "address" : { "building" : "1007", "coord" : [ -73.856077, 40.848447 ], "street" : "Morris Park Ave", "zipcode" : "10462" }, "borough" : "Bronx", "cuisine" : "Bakery", "grades" : [ { "date" : ISODate("2014-03-03T00:00:00Z"), "grade" : "A", "score" : 2 }, { "date" : ISODate("2013-09-11T00:00:00Z"), "grade" : "A", "score" : 6 }, { "date" : ISODate("2013-01-24T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2011-11-23T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2011-03-10T00:00:00Z"), "grade" : "B", "score" : 14 } ], "name" : "Morris Park Bake Shop", "restaurant_id" : "30075445" }
{ "_id" : ObjectId("5bcaca3073c219978a694834"), "address" : { "building" : "469", "coord" : [ -73.961704, 40.662942 ], "street" : "Flatbush Avenue", "zipcode" : "11225" }, "borough" : "Brooklyn", "cuisine" : "Hamburgers", "grades" : [ { "date" : ISODate("2014-12-30T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2014-07-01T00:00:00Z"), "grade" : "B", "score" : 23 }, { "date" : ISODate("2013-04-30T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2012-05-08T00:00:00Z"), "grade" : "A", "score" : 12 } ], "name" : "Wendy'S", "restaurant_id" : "30112340" }
{ "_id" : ObjectId("5bcaca3073c219978a694835"), "address" : { "building" : "351", "coord" : [ -73.98513559999999, 40.7676919 ], "street" : "West 57 Street", "zipcode" : "10019" }, "borough" : "Manhattan", "cuisine" : "Irish", "grades" : [ { "date" : ISODate("2014-09-06T00:00:00Z"), "grade" : "A", "score" : 2 }, { "date" : ISODate("2013-07-22T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2012-07-31T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2011-12-29T00:00:00Z"), "grade" : "A", "score" : 12 } ], "name" : "Dj Reynolds Pub And Restaurant", "restaurant_id" : "30191841" }
{ "_id" : ObjectId("5bcaca3073c219978a694836"), "address" : { "building" : "2780", "coord" : [ -73.98241999999999, 40.579505 ], "street" : "Stillwell Avenue", "zipcode" : "11224" }, "borough" : "Brooklyn", "cuisine" : "American", "grades" : [ { "date" : ISODate("2014-06-10T00:00:00Z"), "grade" : "A", "score" : 5 }, { "date" : ISODate("2013-06-05T00:00:00Z"), "grade" : "A", "score" : 7 }, { "date" : ISODate("2012-04-13T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2011-10-12T00:00:00Z"), "grade" : "A", "score" : 12 } ], "name" : "Riviera Caterer", "restaurant_id" : "40356018" }
>
```

## Question2.

쿼리:

```
db.primer.find({},{name:1,borough:1,cuisine:1,restaurant_id:1,_id:0}).limit(5)
```

'project parameter'기능을 사용한 것으로, find의 두 번째 인자에 출력시킬 filed는 1, 출력시키지 않을 field는 0으로 명시함으로써 find 첫 번째 인자의 조건을 만족시키는 document에서 출력시킬 filed를 제한 할 수 있다.

실행화면

```
> db.primer.find({}, {name:1,borough:1,cuisine:1,restaurant_id:1,_id:0}).limit(5)
{ "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" }
{ "borough" : "Brooklyn", "cuisine" : "American", "name" : "Riviera Caterer", "restaurant_id" : "40356018" }
{ "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Tov Kosher Kitchen", "restaurant_id" : "40356068" }
>
```

### Question3.

#### 쿼리:

```
db.primer.find({borough:"Brooklyn"}).limit(3)
```

Db.primer.find의 첫 번째 인자에 조건으로 borough filed가 "Brooklyn"인 조건을 만족하는 document를 추리고, limit(3)을 통해 3개의 document만 출력한다.

#### 실행화면

```
> db.primer.find({borough:"Brooklyn"}).limit(3)
{ "_id" : ObjectId("5bcaca3073c219978a694834"), "address" : { "building" : "469", "coord" : [ -73.961704, 40.662942 ], "street" : "Flatbush Avenue", "zipcode" : "11225" }, "borough" : "Brooklyn", "cuisine" : "Hamburgers", "grades" : [ { "date" : ISODate("2014-12-30T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2014-07-01T00:00:00Z"), "grade" : "B", "score" : 23 }, { "date" : ISODate("2013-04-30T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2012-05-08T00:00:00Z"), "grade" : "A", "score" : 12 } ], "name" : "Wendy'S", "restaurant_id" : "30112340" }
{ "_id" : ObjectId("5bcaca3073c219978a694836"), "address" : { "building" : "2780", "coord" : [ -73.98241999999999, 40.579505 ], "street" : "Stillwell Avenue", "zipcode" : "11224" }, "borough" : "Brooklyn", "cuisine" : "American", "grades" : [ { "date" : ISODate("2014-06-10T00:00:00Z"), "grade" : "A", "score" : 5 }, { "date" : ISODate("2013-06-05T00:00:00Z"), "grade" : "A", "score" : 7 }, { "date" : ISODate("2012-04-13T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2011-10-12T00:00:00Z"), "grade" : "A", "score" : 12 } ], "name" : "Riviera Caterers", "restaurant_id" : "40356018" }
{ "_id" : ObjectId("5bcaca3073c219978a69483a"), "address" : { "building" : "7114", "coord" : [ -73.9068506, 40.6199034 ], "street" : "Avenue U", "zipcode" : "11234" }, "borough" : "Brooklyn", "cuisine" : "Delicatessen", "grades" : [ { "date" : ISODate("2014-05-29T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2014-01-14T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2013-08-03T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2012-07-18T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2012-03-09T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2011-10-14T00:00:00Z"), "grade" : "A", "score" : 9 } ], "name" : "Wilken'S Fine Food", "restaurant_id" : "40356483" }
>
```

### Question4.

#### 쿼리:

```
db.primer.find({borough:"Brooklyn"}).skip(3).limit(3)
```

db.primer.find의 첫 번째 인자에서 borough가 Brooklyn인 것을 고르고, skip(3)을 통해 앞에서부터 3가지를 skip하고 limit(3)을 통해 남은 document중에서 3가지를 뽑아 출력한다.

#### 실행화면

```
> db.primer.find({borough:"Brooklyn"}).skip(3).limit(3)
{ "_id" : ObjectId("5bcaca3073c219978a69483b"), "address" : { "building" : "6409", "coord" : [ -74.00528899999999, 40.628886 ], "street" : "11 Avenue", "zipcode" : "11219" }, "borough" : "Brooklyn", "cuisine" : "American", "grades" : [ { "date" : ISODate("2014-07-18T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2013-07-30T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2013-02-13T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2012-08-16T00:00:00Z"), "grade" : "A", "score" : 2 }, { "date" : ISODate("2011-08-17T00:00:00Z"), "grade" : "A", "score" : 11 } ], "name" : "Regina Caterers", "restaurant_id" : "40356649" }
{ "_id" : ObjectId("5bcaca3073c219978a69483c"), "address" : { "building" : "1839", "coord" : [ -73.9482609, 40.6408271 ], "street" : "Nostrand Avenue", "zipcode" : "11226" }, "borough" : "Brooklyn", "cuisine" : "Ice Cream, Gelato, Yogurt, Ices", "grades" : [ { "date" : ISODate("2014-07-14T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2013-07-10T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2012-07-11T00:00:00Z"), "grade" : "A", "score" : 5 }, { "date" : ISODate("2012-02-23T00:00:00Z"), "grade" : "A", "score" : 8 } ], "name" : "Taste The Tropics Ice Cream", "restaurant_id" : "40356731" }
{ "_id" : ObjectId("5bcaca3073c219978a69483e"), "address" : { "building" : "7715", "coord" : [ -73.9973325, 40.61174889999999 ], "street" : "18 Avenue", "zipcode" : "11214" }, "borough" : "Brooklyn", "cuisine" : "American", "grades" : [ { "date" : ISODate("2014-04-16T00:00:00Z"), "grade" : "A", "score" : 5 }, { "date" : ISODate("2013-04-23T00:00:00Z"), "grade" : "A", "score" : 2 }, { "date" : ISODate("2012-04-24T00:00:00Z"), "grade" : "A", "score" : 5 }, { "date" : ISODate("2011-12-16T00:00:00Z"), "grade" : "A", "score" : 2 } ], "name" : "C & C Catering Service", "restaurant_id" : "40357437" }
>
```

### Question5.

#### 쿼리:

```
db.primer.find({'grades.score':{$gt:90}}).limit(3)
```

Grades는 리스트 형식으로 그 안에 여러 개의 json object를 가진다. Nested Json Object라고 볼 수 있는데, 여기서 객체안에 score중에 90보다 넘는 것을 찾는다는 predicate 기능을 사용해서 score가 90보다

넘는 것들을 찾고 limit(3)을 통해 3개만 출력한다.

(처음에 grades.score:{\$gt:90}이라고 검색을 했지만, 지속적으로 동작이 정상적으로 하지 않았으나 인터넷 searching결과 nested에 대한 검색을 할 때 ‘ ‘ 를 붙여야 한다는 것을 알게 되었다.

### 실행화면

```
> db.primer.find({'grades.score':{$gt:90}}).limit(3)
{ "_id" : ObjectId("5bcaca3073c219978a694991"), "address" : { "building" : "65", "coord" : [ -73.9782725, 40.7624022 ], "street" : "West 54 Street", "zipcode" : "10019" }, "borough" : "Manhattan", "cuisine" : "American", "grades" : [ { "date" : ISODate("2014-08-22T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2014-03-28T00:00:00Z"), "grade" : "C", "score" : 131 }, { "date" : ISODate("2013-09-25T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2013-04-08T00:00:00Z"), "grade" : "B", "score" : 25 }, { "date" : ISODate("2012-10-15T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2011-10-19T00:00:00Z"), "grade" : "A", "score" : 13 } ], "name" : "Murals On 54/Randolphs'S", "restaurant_id" : "40372466" }, { "_id" : ObjectId("5bcaca3073c219978a694a32"), "address" : { "building" : "345", "coord" : [ -73.9864626, 40.7266739 ], "street" : "East 6 Street", "zipcode" : "10003" }, "borough" : "Manhattan", "cuisine" : "Indian", "grades" : [ { "date" : ISODate("2014-09-15T00:00:00Z"), "grade" : "A", "score" : 5 }, { "date" : ISODate("2014-01-14T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2013-05-30T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2013-04-24T00:00:00Z"), "grade" : "P", "score" : 2 }, { "date" : ISODate("2012-10-01T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2012-04-06T00:00:00Z"), "grade" : "C", "score" : 92 }, { "date" : ISODate("2011-11-03T00:00:00Z"), "grade" : "C", "score" : 41 }, { "date" : ISODate("2011-05-02T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2010-06-17T00:00:00Z"), "grade" : "C", "score" : 98 }, { "date" : ISODate("2013-12-12T00:00:00Z"), "grade" : "C", "score" : 32 }, { "date" : ISODate("2013-05-22T00:00:00Z"), "grade" : "B", "score" : 21 }, { "date" : ISODate("2012-05-02T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2011-10-19T00:00:00Z"), "grade" : "A", "score" : 13 } ], "name" : "Bella Napoli", "restaurant_id" : "40393488" }
>
```

### Question6.

#### 쿼리:

```
db.primer.find({'$and':[{'grades.score':{$gt:80}},{'grades.score':{$lt:100}}]}).limit(3)
```

Predicate ‘And’를 이용해서 grades.score(위에서 말한 json object의 score filed)가 80보다 크고 100보다 작은 두 가지의 조건을 모두 만족하는 것을 추리고, 3개를 출력한다.

### 실행화면

```
{ "_id" : ObjectId("5bcaca3073c219978a694a32"), "address" : { "building" : "345", "coord" : [ -73.9864626, 40.7266739 ], "street" : "East 6 Street", "zipcode" : "10003" }, "borough" : "Manhattan", "cuisine" : "Indian", "grades" : [ { "date" : ISODate("2014-09-15T00:00:00Z"), "grade" : "A", "score" : 5 }, { "date" : ISODate("2014-01-14T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2013-05-30T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2013-04-24T00:00:00Z"), "grade" : "P", "score" : 2 }, { "date" : ISODate("2012-10-01T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2012-04-06T00:00:00Z"), "grade" : "C", "score" : 92 }, { "date" : ISODate("2011-11-03T00:00:00Z"), "grade" : "C", "score" : 41 }, { "date" : ISODate("2011-05-02T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2010-06-17T00:00:00Z"), "grade" : "C", "score" : 98 }, { "date" : ISODate("2013-12-12T00:00:00Z"), "grade" : "C", "score" : 32 }, { "date" : ISODate("2013-05-22T00:00:00Z"), "grade" : "B", "score" : 21 }, { "date" : ISODate("2012-05-02T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2011-10-19T00:00:00Z"), "grade" : "A", "score" : 13 } ], "name" : "Bella Napoli", "restaurant_id" : "40393488" }, { "_id" : ObjectId("5bcaca3073c219978a695404"), "address" : { "building" : "", "coord" : [ -74.0163793, 40.7167671 ], "street" : "Hudson River", "zipcode" : "10282" }, "borough" : "Manhattan", "cuisine" : "American", "grades" : [ { "date" : ISODate("2014-06-27T00:00:00Z"), "grade" : "C", "score" : 89 }, { "date" : ISODate("2013-06-06T00:00:00Z"), "grade" : "A", "score" : 6 }, { "date" : ISODate("2012-06-19T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2011-06-19T00:00:00Z"), "grade" : "A", "score" : 13 } ], "name" : "West 79th Street Boat Basin Cafe", "restaurant_id" : "40756344" }
```

### Question7.

#### 쿼리:

```
db.primer.find({'address.coord.0':{$lt:96.31321}}).limit(3)
```

Address의 coord는 배열 형태로서 index를 사용하여 0 번 index 의 값이 -96.31321보다 작은 것을 출력시키는데 limit(3)을 통하여 3개의 document만 출력한다.

### 실행화면

```
> db.primer.find({'address.coord.0':{'$lt:-96.31321}}).limit(3)
{ "_id" : ObjectId("5bcaca3073c219978a694e7c"), "address" : { "building" : "3707", "coord" : [ -101.8945214, 33.5197474 ], "street" : "82 Street", "zipcode" : "11372", "borough" : "Queens", "cuisine" : "American", "grades" : [ { "date" : ISODate("2014-06-04T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2013-11-07T00:00:00Z"), "grade" : "B", "score" : 19 }, { "date" : ISODate("2013-05-17T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2012-08-29T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2012-04-03T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2011-11-16T00:00:00Z"), "grade" : "A", "score" : 7 } ] }, "name" : "Burger King", "restaurant_id" : "40534067" }
{ "_id" : ObjectId("5bcaca3073c219978a6951e8"), "address" : { "building" : "15259", "coord" : [ -119.6368672, 36.2504996 ], "street" : "10 Avenue", "zipcode" : "11357", "borough" : "Queens", "cuisine" : "Italian", "grades" : [ { "date" : ISODate("2014-09-04T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2014-03-26T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2013-03-04T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2012-09-27T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2012-04-20T00:00:00Z"), "grade" : "A", "score" : 7 }, { "date" : ISODate("2011-11-23T00:00:00Z"), "grade" : "C", "score" : 34 }, { "date" : ISODate("2011-03-07T00:00:00Z"), "grade" : "C", "score" : 41 } ] }, "name" : "Cascario's", "restaurant_id" : "40668681" }
{ "_id" : ObjectId("5bcaca3073c219978a69568f"), "address" : { "building" : "60", "coord" : [ -111.9975205, 42.0970258 ], "street" : "West Side Highway", "zipcode" : "10006", "borough" : "Manhattan", "cuisine" : "Japanese", "grades" : [ { "date" : ISODate("2014-09-15T00:00:00Z"), "grade" : "A", "score" : 5 }, { "date" : ISODate("2014-03-26T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2013-06-17T00:00:00Z"), "grade" : "C", "score" : 98 }, { "date" : ISODate("2013-12-12T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2011-07-27T00:00:00Z"), "grade" : "A", "score" : 2 } ] }, "name" : "Sports Center At Chelsea Piers (Sushi Bar)", "restaurant_id" : "40882356" }
>
```

## Question8.

```
쿼리: db.primer.find({$and:[{'cuisine':{'$ne:'American'}},
{'address.coord.0':{'$lt:-64.32932}},
{'grades.score':{'$gt:80}}]}).limit(3)
```

\$and predicate안에 cuisine이 American이 아닌 것, score가 80보다 큰 것, latitude가 -64.32932보다 작은 것을 모두 만족하는 document 3가지를 출력하도록 한다.

## 실행화면

```
> db.primer.find({$and:[{'cuisine':{'$ne:'American'}},{'address.coord.0':{'$lt:-64.32932}},{'grades.score':{'$gt:80}}]}).limit(3)
{ "_id" : ObjectId("5bcaca3073c219978a694a32"), "address" : { "building" : "345", "coord" : [ -73.9864626, 40.7266739 ], "street" : "East 6 Street", "zipcode" : "10003", "borough" : "Manhattan", "cuisine" : "Indian", "grades" : [ { "date" : ISODate("2014-09-15T00:00:00Z"), "grade" : "A", "score" : 5 }, { "date" : ISODate("2014-01-14T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2013-05-30T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2013-04-24T00:00:00Z"), "grade" : "B", "score" : 2 }, { "date" : ISODate("2012-10-01T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2012-04-06T00:00:00Z"), "grade" : "C", "score" : 92 }, { "date" : ISODate("2011-11-03T00:00:00Z"), "grade" : "C", "score" : 41 } ] }, "name" : "Gandhi", "restaurant_id" : "40381295" }
{ "_id" : ObjectId("5bcaca3073c219978a694b95"), "address" : { "building" : "130", "coord" : [ -73.984758, 40.7457939 ], "street" : "Madison Avenue", "zipcode" : "10016", "borough" : "Manhattan", "cuisine" : "Pizza/Italian", "grades" : [ { "date" : ISODate("2014-12-24T00:00:00Z"), "grade" : "A", "score" : 31 }, { "date" : ISODate("2014-06-17T00:00:00Z"), "grade" : "C", "score" : 98 }, { "date" : ISODate("2013-12-12T00:00:00Z"), "grade" : "A", "score" : 32 }, { "date" : ISODate("2013-05-22T00:00:00Z"), "grade" : "B", "score" : 21 }, { "date" : ISODate("2012-05-02T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2011-11-23T00:00:00Z"), "grade" : "C", "score" : 41 } ] }, "name" : "BellaNapoli", "restaurant_id" : "40393488" }
{ "_id" : ObjectId("5bcaca3073c219978a695942"), "address" : { "building" : "7705", "coord" : [ -73.8878704, 40.7435875 ], "street" : "Woodside Avenue", "zipcode" : "11373", "borough" : "Queens", "cuisine" : "Thai", "grades" : [ { "date" : ISODate("2014-05-12T00:00:00Z"), "grade" : "B", "score" : 14 }, { "date" : ISODate("2013-11-20T00:00:00Z"), "grade" : "C", "score" : 84 }, { "date" : ISODate("2012-09-18T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2012-05-03T00:00:00Z"), "grade" : "B", "score" : 23 } ] }, "name" : "Spicy Shallot", "restaurant_id" : "40979431" }
>
```

## Question9.

## 쿼리:

```
db.primer.aggregate([ {$match:{'cuisine':{'$ne:'American'}}}, {$match:{'address.coord.0':{'$lt:-64.32932}}}, {$match:{'grades.score':{'$gt:80}}}, {$limit:3}])
```

AND OPERATION을 사용하지 않고서 여러 개 조건을 만족하는 쿼리를 사용하려면, AGGREGATE를 사용해야 한다. 또한 이후, 출력 개수를 정하려면, FIND에서는 LIMIT(3) 로 개수를 제한했지만, AGGREGATE에서는 그 안에 \$LIMIT을 통해 파이프라인에서의 다음 단계로 넘어갈 OUTPUT 개수를 제한해야 한다.

## 실행화면

```
> db.primer.aggregate([{$match:{$cuisine:{$ne:'American'}}},{$match:{$address.coord.0:{$lt:-64.32392}}},{$match:{$grades.score:{$gt:80}}},{$limit:3}])
{ "id" : ObjectId("5bcaca3073c219978a694a32"), "address" : { "building" : "345", "coord" : [ -73.9864626, 40.7266739 ], "street" : "East 6 Street", "zipcode" : "10003", "borough" : "Manhattan", "cuisine" : "Indian", "grades" : [ { "date" : ISODate("2014-09-15T00:00:00Z"), "grade" : "A", "score" : 5 }, { "date" : ISODate("2014-01-14T00:00:00Z"), "grade" : "A", "score" : 8 }, { "date" : ISODate("2013-05-30T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2013-04-24T00:00:00Z"), "grade" : "B", "score" : 2 }, { "date" : ISODate("2012-10-01T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2012-04-06T00:00:00Z"), "grade" : "C", "score" : 92 }, { "date" : ISODate("2011-11-03T00:00:00Z"), "grade" : "C", "score" : 41 } ], "name" : "Gandhi", "restaurant_id" : "40381295" }
{ "id" : ObjectId("5bcaca3073c219978a694b95"), "address" : { "building" : "130", "coord" : [ -73.984758, 40.7457939 ], "street" : "Madison Avenue", "zipcode" : "10016", "borough" : "Manhattan", "cuisine" : "Pizza/Italian", "grades" : [ { "date" : ISODate("2014-12-24T00:00:00Z"), "grade" : "Z", "score" : 31 }, { "date" : ISODate("2014-06-17T00:00:00Z"), "grade" : "C", "score" : 98 }, { "date" : ISODate("2013-12-12T00:00:00Z"), "grade" : "C", "score" : 32 }, { "date" : ISODate("2013-05-22T00:00:00Z"), "grade" : "B", "score" : 21 }, { "date" : ISODate("2012-05-02T00:00:00Z"), "grade" : "A", "score" : 11 } ], "name" : "BellaNapolit", "restaurant_id" : "40393488" }
{ "id" : ObjectId("5bcaca3073c219978a695942"), "address" : { "building" : "7705", "coord" : [ -73.8878704, 40.7435875 ], "street" : "Woodside Avenue", "zipcode" : "11373", "borough" : "Queens", "cuisine" : "Thai", "grades" : [ { "date" : ISODate("2014-05-12T00:00:00Z"), "grade" : "B", "score" : 14 }, { "date" : ISODate("2013-11-20T00:00:00Z"), "grade" : "C", "score" : 84 }, { "date" : ISODate("2012-09-18T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2012-05-03T00:00:00Z"), "grade" : "B", "score" : 23 } ], "name" : "Spicy Shallot", "restaurant_id" : "40979431" }
>
```

## Question10.

### 쿼리:

```
db.primer.aggregate([{$match:{$cuisine:{$ne:'American'}}},
{$match:{$grades.grade:'A'}},
{$match:{$borough:{$ne:'Manhattan'}}},
{$sort:{$cuisine:-1}},
{$limit:3}])
```

ARREGATE를 이용해, CUISINE, BOROUGH, GRADE POINT관련 조건을 만족시키고, \$SORT를 통해 CUISINE을 통한 역 정렬, \$LIMIT을 통해 3가지만 출력했다.

### 실행화면

```
> db.primer.aggregate([{$match:{$cuisine:{$ne:'American'}}},{$match:{$grades.grade:'A'}},{$match:{$borough:{$ne:'Manhattan'}}},{$sort:{$cuisine:-1}},{$limit:3}])
{ "id" : ObjectId("5bcaca3073c219978a695af9"), "address" : { "building" : "156-32", "coord" : [ -73.8087286, 40.7639199 ], "street" : "Northern Boulevard", "zipcode" : "11354", "borough" : "Queens", "cuisine" : "Vietnamese/Cambodian/Malaysia", "grades" : [ { "date" : ISODate("2014-09-11T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2014-04-08T00:00:00Z"), "grade" : "A", "score" : 11 }, { "date" : ISODate("2013-10-07T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2012-09-13T00:00:00Z"), "grade" : "A", "score" : 2 }, { "date" : ISODate("2012-04-26T00:00:00Z"), "grade" : "A", "score" : 7 }, { "date" : ISODate("2011-11-09T00:00:00Z"), "grade" : "A", "score" : 10 } ], "name" : "Mekong Thai And Vietnamese Restaurant", "restaurant_id" : "41022628" }
{ "id" : ObjectId("5bcaca3073c219978a6958b8"), "address" : { "building" : "4222", "coord" : [ -73.9996977, 40.6452763 ], "street" : "8 Avenue", "zipcode" : "11232", "borough" : "Brooklyn", "cuisine" : "Vietnamese/Cambodian/Malaysia", "grades" : [ { "date" : ISODate("2014-08-19T00:00:00Z"), "grade" : "B", "score" : 17 }, { "date" : ISODate("2014-03-07T00:00:00Z"), "grade" : "B", "score" : 17 }, { "date" : ISODate("2013-02-01T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2012-01-26T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2011-12-15T00:00:00Z"), "grade" : "B", "score" : 9 } ], "name" : "Ba Xuyen", "restaurant_id" : "40959591" }
{ "id" : ObjectId("5bcaca3073c219978a694ff9"), "address" : { "building" : "8278", "coord" : [ -73.88143509999999, 40.7412552 ], "street" : "Broadway", "zipcode" : "11373", "borough" : "Queens", "cuisine" : "Vietnamese/Cambodian/Malaysia", "grades" : [ { "date" : ISODate("2014-06-12T00:00:00Z"), "grade" : "B", "score" : 21 }, { "date" : ISODate("2013-05-20T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2012-12-26T00:00:00Z"), "grade" : "A", "score" : 10 }, { "date" : ISODate("2012-12-03T00:00:00Z"), "grade" : "F", "score" : 5 }, { "date" : ISODate("2012-05-04T00:00:00Z"), "grade" : "B", "score" : 27 } ], "name" : "Pho Bac Vietnamese Seafood Cuisine", "restaurant_id" : "40578058" }
>
```

## Question11.

### 쿼리:

```
db.primer.find({'borough':'Manhattan',
$or:[{'cuisine':'Chinese'},{'cuisine':'American'}]}).limit(3)
```

Find에서 borough가 manhattan이고, cuisine이 Chinese 또는 american을 포함하는 것을 고르고, limit 통해 3가지만 출력했다. 배열이라면, EQUAL연산에서 여러 개 요소와 비교한다면 해당 요소와 순서와 내용이 같아야 하지만, 한 개와 비교한다면 포함만 하면 True를 Return한다는 점을 이용했다.

### 실행화면

```
> db.primer.find({'borough':'Manhattan', $or:[{'cuisine':'Chinese'},{'cuisine':'American'}]}).limit(3)
{ "_id" : ObjectId("5bcaca3073c219978a694840"), "address" : { "building" : "1", "coord" : [ -73.96926909999999, 40.7685235 ], "street" : "East 66 Street", "zipcode" : "10065" }, "borough" : "Manhattan", "cuisine" : "American", "grades" : [ { "date" : ISODate("2014-05-07T00:00:00Z"), "grade" : "A", "score" : 3 }, { "date" : ISODate("2013-05-03T00:00:00Z"), "grade" : "A", "score" : 4 }, { "date" : ISODate("2012-04-30T00:00:00Z"), "grade" : "A", "score" : 6 }, { "date" : ISODate("2011-12-27T00:00:00Z"), "grade" : "A", "score" : 0 } ], "name" : "1 East 66th Street Kitchen", "restaurant_id" : "40359480" }
{ "_id" : ObjectId("5bcaca3073c219978a694845"), "address" : { "building" : "522", "coord" : [ -73.95171, 40.767461 ], "street" : "East 74 Street", "zipcode" : "10021" }, "borough" : "Manhattan", "cuisine" : "American", "grades" : [ { "date" : ISODate("2014-09-02T00:00:00Z"), "grade" : "A", "score" : 12 }, { "date" : ISODate("2013-12-19T00:00:00Z"), "grade" : "B", "score" : 16 }, { "date" : ISODate("2013-05-28T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2012-12-07T00:00:00Z"), "grade" : "A", "score" : 13 }, { "date" : ISODate("2012-03-29T00:00:00Z"), "grade" : "A", "score" : 11 } ], "name" : "Glorious Food", "restaurant_id" : "40361521" }
{ "_id" : ObjectId("5bcaca3073c219978a69484b"), "address" : { "building" : "730", "coord" : [ -73.96805719999999, 40.7925587 ], "street" : "Columbus Avenue", "zipcode" : "10025" }, "borough" : "Manhattan", "cuisine" : "American", "grades" : [ { "date" : ISODate("2014-09-12T00:00:00Z"), "grade" : "B", "score" : 26 }, { "date" : ISODate("2013-08-28T00:00:00Z"), "grade" : "A", "score" : 9 }, { "date" : ISODate("2013-03-25T00:00:00Z"), "grade" : "B", "score" : 20 }, { "date" : ISODate("2012-02-14T00:00:00Z"), "grade" : "A", "score" : 12 } ], "name" : "P & S Deli Grocery", "restaurant_id" : "40362264" }
>
```

## Question12.

**쿼리:**

```
db.primer.find({'borough':{$in:['StatenIsland','Queens','Bronx','Brooklyn']}},
{'restaurant_id:1,name:1,cuisine:1,borough:1}).limit(5)
```

\$IN을 통해서 BOROUGH가 속할 수 있는 내용들을 적고, 그 안에 속한다면, 해당 document를 limit연산자 통해 5가지만 출력시킨다. 이 때, Find의 두 번째 인자에 출력시킬 내용들을 1,0으로 표현한다.1이면 출력, 0이면 출력하지 않는 문법에 따른다.(projection)

**실행화면**

```
> db.primer.find({'borough':{$in:['Staten Island','Queens','Bronx','Brooklyn']}},{restaurant_id:1,name:1,cuisine:1,borough:1}).limit(5)
{ "_id" : ObjectId("5bcaca3073c219978a694833"), "borough" : "Bronx", "cuisine" : "Bakery", "name" : "Morris Park Bake Shop", "restaurant_id" : "30075445" }
{ "_id" : ObjectId("5bcaca3073c219978a694834"), "borough" : "Brooklyn", "cuisine" : "Hamburgers", "name" : "Wendy'S", "restaurant_id" : "30112340" }
{ "_id" : ObjectId("5bcaca3073c219978a694836"), "borough" : "Brooklyn", "cuisine" : "American", "name" : "Riviera Caterer", "restaurant_id" : "40356018" }
{ "_id" : ObjectId("5bcaca3073c219978a694837"), "borough" : "Queens", "cuisine" : "Jewish/Kosher", "name" : "Tov Kosher Kitchen", "restaurant_id" : "40356068" }
{ "_id" : ObjectId("5bcaca3073c219978a694838"), "borough" : "Queens", "cuisine" : "American", "name" : "Brunos On The Boulevard", "restaurant_id" : "40356151" }
>
```

### Question13.

쿼리:

```
db.primer.aggregate([{$group:{_id:'$borough','number of restaurant':{$sum:1}}}]])
```

Group연산을 통해, borough를 ID로 즉, borough기준으로 해당 borough마다 묶은 다음에 restaurant number라는 필드를 만들어, 그룹별로 묶은 document들의 개수를 합한 것을 return한다. 여기서 이게 restaurant의 개수가 되는 것은, document마다 borough를 한 개씩 갖고, 또한 document에서 restaurant\_id(restaurant각각의 이름)을 하나씩 갖는 일대일 관계라서 restaurant\_id의 합이 총 개수와 같다고 판단할 수 있다.

실행화면

```
> db.primer.aggregate([{$group:{_id:'$borough','number of restaurant':{$sum:1}}}]])
{ "_id" : "Missing", "number of restaurant" : 51 }
{ "_id" : "Staten Island", "number of restaurant" : 969 }
{ "_id" : "Brooklyn", "number of restaurant" : 6086 }
{ "_id" : "Bronx", "number of restaurant" : 2338 }
{ "_id" : "Queens", "number of restaurant" : 5656 }
{ "_id" : "Manhattan", "number of restaurant" : 10259 }
>
```

### Question14.

쿼리:

```
db.primer.aggregate([{$match:{'borough':'Queens','cuisine':'Brazilian'}}, {$group:{_id:'$address.zipcode','count':{$sum:1}}}]])
```

Aggregate연산을 통해, borough가 queens인 것을 고르고 cuisine이 brazilian에 해당하는 것을 다음 파이프라인 단계의 input으로 넘긴다. 그리고 address.zipcode마다 group id를 부여하여 거기에 해당하는 레스토랑의 개수를 더한다. 그 후 출력한다.

실행화면

```
> db.primer.aggregate([{$match:{'borough':'Queens','cuisine':'Brazilian'}}, {$group:{_id:'$address.zipcode','count':{$sum:1}}}]])
{ "_id" : "11377", "count" : 1 }
{ "_id" : "11103", "count" : 1 }
{ "_id" : "11106", "count" : 3 }
{ "_id" : "11368", "count" : 1 }
{ "_id" : "11101", "count" : 2 }
>
```

### Question15.

쿼리:

```
db.primer.aggregate([{$match: {'cuisine': 'American'}},  
{$group: { '_id': '$borough', 'number of American  
restaurants': {$sum: 1} }}, {$sort: {'number of American  
restaurants': -1}}, {$project: {borough: 1, 'number of American  
restaurants': 1}}])
```

Aggregate로 우선, cuisine이 American인 것을 검색한다. 그리고 그 결과를 다음 파이프라인 group의 입력으로 준다. group에서는 borough를 기준으로 각각의 레스토랑을 세고, number of American restaurant라는 필드를 만들어 American cuisine을 가지고 있는 레스토랑들을 borough별로 그룹화한다. 그 후, 해당 필드를 기준으로 descending으로 sort한 후, project 연산을 통해 최종 출력에 포함시킬 결과물을 정한다

실행화면

```
> db.primer.aggregate([{$match: {'cuisine': 'American'}}, {$group: { '_id': '$borough', 'number of American restaurants': {$sum: 1} }}, {$sort: {'number of American restaurants': -1}}, {$project: {borough: 1, 'number of American restaurants': 1}}])  
{ "_id": "Manhattan", "number of American restaurants": 3205 }  
{ "_id": "Brooklyn", "number of American restaurants": 1273 }  
{ "_id": "Queens", "number of American restaurants": 1040 }  
{ "_id": "Bronx", "number of American restaurants": 411 }  
{ "_id": "Staten Island", "number of American restaurants": 244 }  
{ "_id": "Missing", "number of American restaurants": 10 }  
>
```