

AGGREGATION

1) Create database Restaurant.

```
> use Restaurants
```

```
'switched to db Restaurant'
```

2) Create collection Hotel.

```
> db.createCollection("Hotel")
```

```
{ ok: 1 }
```

3) Insert 10 documents with above mentioned structure.

```
> db.Hotel.insertMany([
  {Hotel_id:11,Hotel_name:"Ryokan",Type:"Hotel+Restaurant",
  Ratings:5,
  Address:{Area:"Shibuya",City:"Tokyo",Pincode:"4411000"},
  Rooms:[{Room_no:402,Type:"Twin
  Sharing",Price:"400029"},{Room_no:501,Type:"Presidential
  Suite",Price:"800000"},{Room_no:100,Type:"Non AC",Price:"90000"}],
  Cuisines:["Indian","Italian","Chinese","Spanish"],
  likes:51000},{Hotel_id:12,
  Hotel_name:"Sega",
  Type:"Hotel+Restaurant",
  Ratings:5,
  Address:{Area:"Nori",City:"Osaka",Pincode:"4511000"},
  Rooms:[{Room_no:402,Type:"Twin
  Sharing",Price:"329332"},{Room_no:501,Type:"Presidential
  Suite",Price:"500000"},{Room_no:100,Type:"Non AC",Price:"90000"}],
  Cuisines:["Indian","Italian","Chinese","Spanish"],
  likes:52000}
])
```

```
{ acknowledged: true,
  insertedIds:
    { '0': ObjectId("62fa15f916b69a2117c5c78f"),
      '1': ObjectId("62fa15f916b69a2117c5c790") } }
```

4) Display all Hotel information.

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

```
{ _id: ObjectId("62dcde4f11d219dd18a3fa8f"),
```

```
Hotel_id: 1,
Hotel_name: 'Sheraton',
Type: 'Hotel+Restaurant',
Ratings: 9,
Address: { Area: 'KP', City: 'Pune', Pincode: 411001 },
Room:
[ { Room_no: 101, Type: 'Deluxe', Price: 40000 },
  { Room_no: 102, Type: 'Suite', Price: 40000 } ],
Cuisines: [ 'Indian', 'Chinese', 'Italian' ],
likes: 400000 }
{ _id: ObjectId("62dce02011d219dd18a3fa90"),
Hotel_id: 2,
Hotel_name: 'Lemon Tree',
Type: 'Hotel+Restaurant',
Ratings: 4,
Address: { Area: 'Dattawadi', City: 'Pune', Pincode: 411009 },
Room:
[ { Room_no: 1000, Type: 'Deluxe', Price: 20000 },
  { Room_no: 1001, Type: 'Suite', Price: 20000 } ],
Cuisines: [ 'Indian', 'Italian' ],
likes: 2000 }
{ _id: ObjectId("62dce02011d219dd18a3fa91"),
Hotel_id: 3,
Hotel_name: 'Taj',
Type: 'Hotel+Restaurant',
Ratings: 10,
Address: { Area: 'Worli', City: 'Mumbai', Pincode: 400001 },
Room:
[ { Room_no: 101, Type: 'Deluxe', Price: 100000 },
  { Room_no: 102, Type: 'Suite', Price: 200000 } ],
Cuisines: [ 'Indian', 'Chinese', 'Italian' ],
likes: 1000000 }
{ _id: ObjectId("62dce14411d219dd18a3fa92"),
Hotel_id: 4,
Hotel_name: 'Novotel',
Type: 'Hotel+Restaurant',
Ratings: 6,
Address: { Area: 'KP', City: 'Pune', Pincode: 411001 },
Room:
[ { Room_no: 1000, Type: 'Deluxe', Price: 30000 },
  { Room_no: 1001, Type: 'Treehouse', Price: 20000 } ],
Cuisine: [ 'Indian', 'Chinese' ],
likes: 2000 }
{ _id: ObjectId("62dce14411d219dd18a3fa93"),
Hotel_id: 5,
Hotel_name: 'Ibis',
Type: 'Hotel+Restaurant',
Ratings: 5,
Address: { Area: 'Bandra', City: 'Mumbai', Pincode: 400002 },
Room:
```

```
[ { Room_no: 101, Type: 'Deluxe', Price: 60000 },
  { Room_no: 102, Type: 'Sea-facing', Price: 80000 } ],
Cuisines: [ 'Indian' ],
likes: 50000 }
{ _id: ObjectId("62dce27411d219dd18a3fa94"),
  Hotel_id: 6,
  Hotel_name: 'Conrad',
  Type: 'Hotel+Restaurant',
  Ratings: 8,
  Address: { Area: 'Alandur', City: 'Chennai', Pincode: 600001 },
  Room:
    [ { Room_no: 10, Type: 'Suite', Price: 80000 },
      { Room_no: 11, Type: 'Sea-facing', Price: 85000 } ],
    Cuisines: [ 'Indian', 'Italian' ],
    likes: 2000 }
{ _id: ObjectId("62dce27411d219dd18a3fa95"),
  Hotel_id: 7,
  Hotel_name: 'Vivanta',
  Type: 'Hotel+Restaurant',
  Ratings: 5,
  Address: { Area: 'Goregaon', City: 'Mumbai', Pincode: 400003 },
  Room:
    [ { Room_no: 111, Type: 'Deluxe', Price: 60000 },
      { Room_no: 112, Type: 'Sea-facing', Price: 80000 } ],
    Cuisine: [ 'Indian', 'Italian' ],
    likes: 6000 }
{ _id: ObjectId("62dce49811d219dd18a3fa96"),
  Hotel_id: 8,
  Hotel_name: 'JW Marriott',
  Type: 'Hotel+Restaurant',
  Ratings: 8,
  Address: { Area: 'Anna Nagar', City: 'Chennai', Pincode: 600002 },
  Room:
    [ { Room_no: 1001, Type: 'Suite', Price: 89000 },
      { Room_no: 11, Type: 'Sea-facing', Price: 100000 } ],
    Cuisines: [ 'Indian', 'Chinese', 'Italian' ],
    likes: 67000 }
{ _id: ObjectId("62dce49811d219dd18a3fa97"),
  Hotel_id: 9,
  Hotel_name: 'Oberoi',
  Type: 'Hotel+Restaurant',
  Ratings: 8,
  Address: { Area: 'Baner', City: 'Pune', Pincode: 411045 },
  Room:
    [ { Room_no: 112, Type: 'Deluxe', Price: 67000 },
      { Room_no: 114, Type: 'Suite', Price: 85600 } ],
    Cuisines: [ 'Indian', 'Italian' ],
    likes: 89000 }
{ _id: ObjectId("62dce72d11d219dd18a3fa98"),
  Hotel_id: 10,
```

```

Hotel_name: 'Ginger',
Type: 'Hotel+Restaurant',
Ratings: 3,
Address: { Area: 'Malad', City: 'Mumbai', Pincode: 400009 },
Room: [
  { Room_no: 118, Type: 'Suite', Price: 20000 },
  { Room_no: 119, Type: 'Sea-facing', Price: 30000 } ],
Cuisine: [ 'Indian', 'Chinese', 'Italian' ],
likes: 2000 }

```

5) Display no of rooms in each hotel

```

> db.Hotel.aggregate({$project:{"No of
rooms":{$size:"$Rooms"},_id:0,"Hotel_name":1}}).pretty()
{ Hotel_name: 'Sheraton', 'No of rooms': 2 }
{ Hotel_name: 'Lemon Tree', 'No of rooms': 2 }
{ Hotel_name: 'Taj', 'No of rooms': 2 }
{ Hotel_name: 'Novotel', 'No of rooms': 2 }
{ Hotel_name: 'Ibis', 'No of rooms': 2 }
{ Hotel_name: 'Conrad', 'No of rooms': 2 }
{ Hotel_name: 'Vivanta', 'No of rooms': 2 }
{ Hotel_name: 'JW Marriott', 'No of rooms': 2 }
{ Hotel_name: 'Oberoi', 'No of rooms': 2 }
{ Hotel_name: 'Ginger', 'No of rooms': 2 }

```

6) Compute the top five hotels

```

>db.Hotel.aggregate([{$group: {_id:"$Hotel_name",no_of_likes:{$sum:"$likes"}}},{ "$sort":{"no_of_likes":-1}},{ $limit:5}])
{ _id: 'Taj', no_of_likes: 10000000 }
{ _id: 'Sheraton', no_of_likes: 400000 }
{ _id: 'Oberoi', no_of_likes: 89000 }
{ _id: 'JW Marriott', no_of_likes: 67000 }
{ _id: 'Ibis', no_of_likes: 50000 }

```

7) Return hotels having likes above 2000

```

>db.Hotel.aggregate([{$group: {_id:"$Hotel_name",Total_likes:{$sum:"$likes"}},{ $match:{Total_likes:{$gt:2000}}}]])
{ _id: 'Taj', Total_likes: 10000000 }
{ _id: 'Ibis', Total_likes: 50000 }
{ _id: 'Vivanta', Total_likes: 6000 }
{ _id: 'Sheraton', Total_likes: 400000 }
{ _id: 'JW Marriott', Total_likes: 67000 }
{ _id: 'Oberoi', Total_likes: 89000 }

```

8) Return the Five Most Common Cuisines

```
>db.Hotel.aggregate([{$unwind:"$Cuisines"},{$group : { _id : '$Cuisines', count:{$sum : 1}}},{$sort:"-count"}])
```

```
{ _id: 'Indian', count: 10 }
{ _id: 'Italian', count: 8 }
{ _id: 'Chinese', count: 5 }
```

9) Return all prices of room in different hotel of type 'Deluxe'.

```
>db.Hotel.aggregate([{$unwind:"$Rooms"},{$match:{"Rooms.Type":"Deluxe"}},{$project:{_id:0,"Hotel_name":1,"Rooms.Type":1,"Rooms.Price":1}}])
```

```
{ Hotel_name: 'Sheraton',
  Room: { Type: 'Deluxe', Price: 40000 } }
{ Hotel_name: 'Lemon Tree',
  Room: { Type: 'Deluxe', Price: 20000 } }
{ Hotel_name: 'Taj', Rooms: { Type: 'Deluxe', Price: 100000 } }
{ Hotel_name: 'Novotel',
  Room: { Type: 'Deluxe', Price: 30000 } }
{ Hotel_name: 'Ibis', Rooms: { Type: 'Deluxe', Price: 60000 } }
{ Hotel_name: 'Vivanta',
  Room: { Type: 'Deluxe', Price: 60000 } }
{ Hotel_name: 'Oberoi',
  Room: { Type: 'Deluxe', Price: 67000 } }
```

10) Get the total count of hotels having ratings '5 star'

```
> db.Hotel.aggregate([{$match:{"Ratings": 5}},{$count: "no of hotels with rating 5"}])
```

```
{ 'no of hotels with rating 5': 2 }
```

11) Display the count of hotels from 'Pune' city.

```
> db.Hotel.aggregate([{$match:{"Address.City": "Pune"}},{$count: "No of hotels in Pune: "}]
```

```
{ 'No of hotels in Pune: ': 4 }
```

INDEXING

1) Import restaurant.json

```
> mongoimport restaurant.json -d hoteldb -c hotel
```

2) Single field index

```
> db.restaurant.createIndex({borough:1});
```

```
'borough_1'
```

```
> db.restaurant.find({"borough":"Bronx"})
```

```
{ _id: ObjectId("631f27543535c9414d4b07dc"),  
  address:  
    { building: '1007',  
      coord: [ -73.856077, 40.848447 ],  
      street: 'Morris Park Ave',  
      zipcode: '10462' },  
    borough: 'Bronx',  
    cuisine: 'Bakery',  
    grades:  
      [ { date: 2014-03-03T00:00:00.000Z, grade: 'A', score: 2 },  
        { date: 2013-09-11T00:00:00.000Z, grade: 'A', score: 6 },  
        { date: 2013-01-24T00:00:00.000Z, grade: 'A', score: 10 },  
        { date: 2011-11-23T00:00:00.000Z, grade: 'A', score: 9 },  
        { date: 2011-03-10T00:00:00.000Z, grade: 'B', score: 14 } ],  
    name: 'Morris Park Bake Shop',  
    restaurant_id: '30075445' }  
{ _id: ObjectId("631f27543535c9414d4b07e6"),  
  address:  
    { building: '2300',  
      coord: [ -73.8786113, 40.8502883 ],  
      street: 'Southern Boulevard',  
      zipcode: '10460' },  
    borough: 'Bronx',  
    cuisine: 'American ',  
    grades:  
      [ { date: 2014-05-28T00:00:00.000Z, grade: 'A', score: 11 },  
        { date: 2013-06-19T00:00:00.000Z, grade: 'A', score: 4 },  
        { date: 2012-06-15T00:00:00.000Z, grade: 'A', score: 3 } ],  
    name: 'Wild Asia',  
    restaurant_id: '40357217' }  
{ _id: ObjectId("631f27543535c9414d4b07fb"),  
  address:  
    { building: '1006',  
      coord: [ -73.84856870000002, 40.8903781 ],  
      street: 'East 233 Street',  
      zipcode: '10466' },  
    borough: 'Bronx',  
    cuisine: 'Ice Cream, Gelato, Yogurt, Ices',  
    grades:  
      [ { date: 2014-04-24T00:00:00.000Z, grade: 'A', score: 10 },  
        { date: 2013-09-05T00:00:00.000Z, grade: 'A', score: 10 },
```

```

{ date: 2013-02-21T00:00:00.000Z, grade: 'A', score: 9 },
{ date: 2012-07-03T00:00:00.000Z, grade: 'A', score: 11 },
{ date: 2011-07-11T00:00:00.000Z, grade: 'A', score: 5 } ],
name: 'Carvel Ice Cream',
restaurant_id: '40363093' }
{ _id: ObjectId("631f27543535c9414d4b07ff"),
address:
{ building: '1236',
coord: [ -73.8893654, 40.81376179999999 ],
street: '238 Spofford Ave',
zipcode: '10474' },
borough: 'Bronx',
cuisine: 'Chinese',
grades:
[ { date: 2013-12-30T00:00:00.000Z, grade: 'A', score: 8 },
{ date: 2013-01-08T00:00:00.000Z, grade: 'A', score: 10 },
{ date: 2012-06-12T00:00:00.000Z, grade: 'B', score: 15 } ],
name: 'Happy Garden',
restaurant_id: '40363289' }
{ _id: ObjectId("631f27543535c9414d4b0811"),
address:
{ building: '277',
coord: [ -73.8941893, 40.8634684 ],
street: 'East Kingsbridge Road',
zipcode: '10458' },
borough: 'Bronx',
cuisine: 'Chinese',
grades:
[ { date: 2014-03-03T00:00:00.000Z, grade: 'A', score: 10 },
{ date: 2013-09-26T00:00:00.000Z, grade: 'A', score: 10 },
{ date: 2013-03-19T00:00:00.000Z, grade: 'A', score: 10 },
{ date: 2012-08-29T00:00:00.000Z, grade: 'A', score: 11 },
{ date: 2011-08-17T00:00:00.000Z, grade: 'A', score: 13 } ],
name: 'Happy Garden',
restaurant_id: '40364296' }
{ _id: ObjectId("631f27543535c9414d4b0819"),
address:
{ building: '658',
coord: [ -73.81363999999999, 40.82941100000001 ],
street: 'Clarence Ave',
zipcode: '10465' },
borough: 'Bronx',
cuisine: 'American ',
grades:
[ { date: 2014-06-21T00:00:00.000Z, grade: 'A', score: 5 },
{ date: 2012-07-11T00:00:00.000Z, grade: 'A', score: 10 } ],
name: 'Manhem Club',
restaurant_id: '40364363' }
{ _id: ObjectId("631f27543535c9414d4b0831"),
address:

```

```

{ building: '2222',
  coord: [ -73.84971759999999, 40.8304811 ],
  street: 'Haviland Avenue',
  zipcode: '10462' },
borough: 'Bronx',
cuisine: 'American ',
grades:
[ { date: 2014-12-18T00:00:00.000Z, grade: 'A', score: 7 },
  { date: 2014-05-01T00:00:00.000Z, grade: 'B', score: 17 },
  { date: 2013-03-14T00:00:00.000Z, grade: 'A', score: 12 },
  { date: 2012-09-20T00:00:00.000Z, grade: 'A', score: 9 },
  { date: 2012-02-08T00:00:00.000Z, grade: 'B', score: 19 } ],
name: 'The New Starling Athletic Club Of The Bronx',
restaurant_id: '40364956' }
{ _id: ObjectId("631f27543535c9414d4b0849"),
  address:
  { building: '72',
    coord: [ -73.92506, 40.8275556 ],
    street: 'East 161 Street',
    zipcode: '10451' },
    borough: 'Bronx',
    cuisine: 'American ',
    grades:
    [ { date: 2014-04-15T00:00:00.000Z, grade: 'A', score: 9 },
      { date: 2013-11-14T00:00:00.000Z, grade: 'A', score: 4 },
      { date: 2013-07-29T00:00:00.000Z, grade: 'A', score: 10 },
      { date: 2012-12-31T00:00:00.000Z, grade: 'B', score: 15 },
      { date: 2012-05-30T00:00:00.000Z, grade: 'A', score: 13 },
      { date: 2012-01-09T00:00:00.000Z, grade: 'A', score: 10 },
      { date: 2011-08-15T00:00:00.000Z, grade: 'C', score: 37 } ],
    name: 'Yankee Tavern',
    restaurant_id: '40365499' }
  { _id: ObjectId("631f27543535c9414d4b085c"),
    address:
    { building: '331',
      coord: [ -73.87786539999999, 40.8724377 ],
      street: 'East 204 Street',
      zipcode: '10467' },
      borough: 'Bronx',
      cuisine: 'Irish',
      grades:
      [ { date: 2014-08-26T00:00:00.000Z, grade: 'A', score: 10 },
        { date: 2014-03-26T00:00:00.000Z, grade: 'B', score: 23 },
        { date: 2013-09-11T00:00:00.000Z, grade: 'A', score: 13 },
        { date: 2012-12-18T00:00:00.000Z, grade: 'B', score: 27 },
        { date: 2011-10-20T00:00:00.000Z, grade: 'A', score: 13 } ],
      name: 'Mcdwyers Pub',
      restaurant_id: '40365893' }
    { _id: ObjectId("631f27543535c9414d4b0875"),
      address:

```



```

{ building: '5820',
  coord: [ -73.9002615, 40.885186 ],
  street: 'Broadway',
  zipcode: '10463' },
borough: 'Bronx',
cuisine: 'American ',
grades:
[ { date: 2014-02-26T00:00:00.000Z, grade: 'A', score: 5 },
  { date: 2013-10-09T00:00:00.000Z, grade: 'B', score: 19 },
  { date: 2013-05-15T00:00:00.000Z, grade: 'A', score: 9 },
  { date: 2012-11-20T00:00:00.000Z, grade: 'B', score: 18 },
  { date: 2011-10-17T00:00:00.000Z, grade: 'A', score: 10 },
  { date: 2011-06-22T00:00:00.000Z, grade: 'C', score: 35 } ],
name: 'The Punch Bowl',
restaurant_id: '40366497' }
{ _id: ObjectId("631f27543535c9414d4b0880"),
  address:
  { building: '21',
    coord: [ -73.9168424, 40.8401362 ],
    street: 'East 170 Street',
    zipcode: '10452' },
    borough: 'Bronx',
    cuisine: 'American ',
    grades:
    [ { date: 2014-12-16T00:00:00.000Z, grade: 'B', score: 22 },
      { date: 2014-08-12T00:00:00.000Z, grade: 'A', score: 7 },
      { date: 2014-03-03T00:00:00.000Z, grade: 'B', score: 22 },
      { date: 2013-08-29T00:00:00.000Z, grade: 'A', score: 12 },
      { date: 2012-08-29T00:00:00.000Z, grade: 'A', score: 10 },
      { date: 2012-02-13T00:00:00.000Z, grade: 'B', score: 16 } ],
    name: 'Munchtime',
    restaurant_id: '40366748' }
  { _id: ObjectId("631f27543535c9414d4b0882"),
    address:
    { building: '4340',
      coord: [ -73.8194559, 40.8899176 ],
      street: 'Boston Road',
      zipcode: '10475' },
      borough: 'Bronx',
      cuisine: 'Pancakes/Waffles',
      grades:
      [ { date: 2014-09-22T00:00:00.000Z, grade: 'A', score: 11 },
        { date: 2014-05-16T00:00:00.000Z, grade: 'A', score: 9 },
        { date: 2013-10-30T00:00:00.000Z, grade: 'A', score: 3 },
        { date: 2013-06-05T00:00:00.000Z, grade: 'B', score: 24 },
        { date: 2012-05-14T00:00:00.000Z, grade: 'A', score: 13 },
        { date: 2012-01-04T00:00:00.000Z, grade: 'A', score: 7 } ],
      name: 'Ihop',
      restaurant_id: '40366833' }
    { _id: ObjectId("631f27543535c9414d4b0887"),

```

```

address:
{ building: '1191',
coord: [ -73.8513114, 40.8316981 ],
street: 'Castle Hill Avenue',
zipcode: '10462' },
borough: 'Bronx',
cuisine: 'Café/Coffee/Tea',
grades:
[ { date: 2014-07-03T00:00:00.000Z, grade: 'A', score: 12 },
{ date: 2013-07-08T00:00:00.000Z, grade: 'A', score: 9 },
{ date: 2013-02-11T00:00:00.000Z, grade: 'A', score: 11 },
{ date: 2012-08-14T00:00:00.000Z, grade: 'B', score: 14 },
{ date: 2011-08-23T00:00:00.000Z, grade: 'A', score: 12 } ],
name: 'Lulu\'S Coffee Shop',
restaurant_id: '40366938' }
{ _id: ObjectId("631f27543535c9414d4b088d"),
address:
{ building: '2894',
coord: [ -73.815175, 40.8137649 ],
street: 'Schurz Ave',
zipcode: '10465' },
borough: 'Bronx',
cuisine: 'American ',
grades:
[ { date: 2014-10-31T00:00:00.000Z, grade: 'B', score: 18 },
{ date: 2014-02-20T00:00:00.000Z, grade: 'A', score: 4 },
{ date: 2013-06-07T00:00:00.000Z, grade: 'A', score: 12 },
{ date: 2012-05-17T00:00:00.000Z, grade: 'A', score: 2 } ],
name: 'Marina Delray',
restaurant_id: '40367161' }
{ _id: ObjectId("631f27543535c9414d4b08b2"),
address:
{ building: '4280',
coord: [ -73.86697, 40.897946999999999 ],
street: 'Katonah Ave',
zipcode: '10470' },
borough: 'Bronx',
cuisine: 'Not Listed/Not Applicable',
grades:
[ { date: 2014-06-04T00:00:00.000Z, grade: 'A', score: 5 },
{ date: 2013-06-07T00:00:00.000Z, grade: 'A', score: 2 },
{ date: 2012-06-27T00:00:00.000Z, grade: 'A', score: 2 },
{ date: 2011-07-13T00:00:00.000Z, grade: 'A', score: 10 } ],
name: 'The Lark\'S Nest',
restaurant_id: '40367946' }
{ _id: ObjectId("631f27543535c9414d4b08b3"),
address:
{ building: '2300',
coord: [ -73.8786113, 40.8502883 ],
street: 'Southern Boulevard',

```

```

zipcode: '10460' },
borough: 'Bronx',
cuisine: 'Café/Coffee/Tea',
grades:
[ { date: 2014-05-21T00:00:00.000Z, grade: 'A', score: 8 },
  { date: 2013-05-29T00:00:00.000Z, grade: 'A', score: 9 },
  { date: 2012-06-18T00:00:00.000Z, grade: 'A', score: 8 } ],
name: 'Terrace Cafe',
restaurant_id: '40368018' }
{ _id: ObjectId("631f27543535c9414d4b08b4"),
address:
{ building: '2300',
coord: [ -73.8786113, 40.8502883 ],
street: 'Southern Boulevard',
zipcode: '10460' },
borough: 'Bronx',
cuisine: 'African',
grades:
[ { date: 2014-05-21T00:00:00.000Z, grade: 'A', score: 8 },
  { date: 2013-06-01T00:00:00.000Z, grade: 'A', score: 6 },
  { date: 2012-06-12T00:00:00.000Z, grade: 'A', score: 0 },
  { date: 2011-06-08T00:00:00.000Z, grade: 'A', score: 0 } ],
name: 'African Terrace',
restaurant_id: '40368021' }
{ _id: ObjectId("631f27543535c9414d4b08b5"),
address:
{ building: '2300',
coord: [ -73.8786113, 40.8502883 ],
street: 'Southern Boulevard',
zipcode: '10460' },
borough: 'Bronx',
cuisine: 'American ',
grades:
[ { date: 2014-05-21T00:00:00.000Z, grade: 'A', score: 2 },
  { date: 2013-05-22T00:00:00.000Z, grade: 'A', score: 11 },
  { date: 2012-05-17T00:00:00.000Z, grade: 'A', score: 12 },
  { date: 2011-06-08T00:00:00.000Z, grade: 'A', score: 5 } ],
name: 'Cool Zone',
restaurant_id: '40368022' }
{ _id: ObjectId("631f27543535c9414d4b08b6"),
address:
{ building: '2300',
coord: [ -73.8786113, 40.8502883 ],
street: 'Southern Boulevard',
zipcode: '10460' },
borough: 'Bronx',
cuisine: 'American ',
grades:
[ { date: 2014-05-21T00:00:00.000Z, grade: 'A', score: 11 },
  { date: 2013-05-29T00:00:00.000Z, grade: 'A', score: 10 },

```

```
{ date: 2012-06-15T00:00:00.000Z, grade: 'A', score: 2 },
{ date: 2011-06-08T00:00:00.000Z, grade: 'A', score: 0 } ],
name: 'Beaver Pond',
restaurant_id: '40368025' }
{ _id: ObjectId('631f27543535c9414d4b08b7'),
address:
{ building: '2300',
coord: [ -73.8786113, 40.8502883 ],
street: 'Southern Boulevard',
zipcode: '10460' },
borough: 'Bronx',
cuisine: 'American ',
grades:
[ { date: 2014-05-21T00:00:00.000Z, grade: 'A', score: 5 },
{ date: 2013-05-28T00:00:00.000Z, grade: 'A', score: 3 },
{ date: 2012-06-18T00:00:00.000Z, grade: 'A', score: 4 },
{ date: 2011-06-07T00:00:00.000Z, grade: 'A', score: 9 } ],
name: 'African Market (Baboon Cafe)',
restaurant_id: '40368026' }
```

Type "it" for more

```
> db.restaurant.getIndexes()
[
{ v: 2, key: { _id: 1 }, name: '_id' },
{ v: 2, key: { borough: 1 }, name: 'borough_1' },
]
```

```
> db.restaurant.explain().find()
{ explainVersion: '1',
queryPlanner:
{ namespace: 'hotel.restaurant',
indexFilterSet: false,
parsedQuery: {},
queryHash: '8B3D4AB8',
planCacheKey: 'D542626C',
maxIndexedOrSolutionsReached: false,
maxIndexedAndSolutionsReached: false,
maxScansToExplodeReached: false,
winningPlan: { stage: 'COLLSCAN', direction: 'forward' },
rejectedPlans: [] },
command: { find: 'restaurant', filter: {}, '$db': 'hotel' },
serverInfo:
{ host: 'LAPTOP-SF40VR81',
port: 27017,
version: '5.0.12',
gitVersion: '79cfcdd83eb6f64e164a588d0daf9bb873328b45' },
serverParameters:
{ internalQueryFacetBufferSizeBytes: 104857600,
internalQueryFacetMaxOutputDocSizeBytes: 104857600,
```

```
internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,  
internalDocumentSourceGroupMaxMemoryBytes: 104857600,  
internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,  
internalQueryProhibitBlockingMergeOnMongoS: 0,  
internalQueryMaxAddToSetBytes: 104857600,  
internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600 },  
ok: 1 }
```

```
> db.restaurant.find({"borough":"Bronx"}).explain("allPlansExecution")  
{ explainVersion: '1',  
  queryPlanner:  
    { namespace: 'hotel.restaurant',  
      indexFilterSet: false,  
      parsedQuery: { borough: { '$eq': 'Bronx' } } },  
      maxIndexedOrSolutionsReached: false,  
      maxIndexedAndSolutionsReached: false,  
      maxScansToExplodeReached: false,  
      winningPlan:  
        { stage: 'FETCH',  
          inputStage:  
            { stage: 'IXSCAN',  
              keyPattern: { borough: 1 },  
              indexName: 'borough_1',  
              isMultiKey: false,  
              multiKeyPaths: { borough: [] },  
              isUnique: false,  
              isSparse: false,  
              isPartial: false,  
              indexVersion: 2,  
              direction: 'forward',  
              indexBounds: { borough: [ ['Bronx', 'Bronx'] ] } } },  
              rejectedPlans: [] },  
              executionStats:  
                { executionSuccess: true,  
                  nReturned: 309,  
                  executionTimeMillis: 37,  
                  totalKeysExamined: 309,  
                  totalDocsExamined: 309,  
                  executionStages:  
                    { stage: 'FETCH',  
                      nReturned: 309,  
                      executionTimeMillisEstimate: 0,  
                      works: 310,  
                      advanced: 309,  
                      needTime: 0,  
                      needYield: 0,  
                      saveState: 1,  
                      restoreState: 1,
```

```
isEOF: 1,
docsExamined: 309,
alreadyHasObj: 0,
inputStage:
{ stage: 'IXSCAN',
nReturned: 309,
executionTimeMillisEstimate: 0,
works: 310,
advanced: 309,
needTime: 0,
needYield: 0,
saveState: 1,
restoreState: 1,
isEOF: 1,
keyPattern: { borough: 1 },
indexName: 'borough_1',
isMultiKey: false,
multiKeyPaths: { borough: [] },
isUnique: false,
isSparse: false,
isPartial: false,
indexVersion: 2,
direction: 'forward',
indexBounds: { borough: [ ['"Bronx"', '"Bronx"'] ] },
keysExamined: 309,
seeks: 1,
dupsTested: 0,
dupsDropped: 0 } },
allPlansExecution: [] },
command:
{ find: 'restaurant',
filter: { borough: 'Bronx' },
'$db': 'hotel' },
serverInfo:
{ host: 'LAPTOP-SF40VR81',
port: 27017,
version: '5.0.12',
gitVersion: '79cfcdd83eb6f64e164a588d0daf9bb873328b45' },
serverParameters:
{ internalQueryFacetBufferSizeBytes: 104857600,
internalQueryFacetMaxOutputDocSizeBytes: 104857600,
internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
internalDocumentSourceGroupMaxMemoryBytes: 104857600,
internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
internalQueryProhibitBlockingMergeOnMongoS: 0,
internalQueryMaxAddToSetBytes: 104857600,
internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600 },
ok: 1 }
```

```
db.restaurant.find({"borough":"Bronx"}).explain("executionStats")
{ explainVersion: '1',
  queryPlanner:
    { namespace: 'hotel.restaurant',
      indexFilterSet: false,
      parsedQuery: { borough: { '$eq': 'Bronx' } },
      maxIndexedOrSolutionsReached: false,
      maxIndexedAndSolutionsReached: false,
      maxScansToExplodeReached: false,
      winningPlan:
        { stage: 'FETCH',
          inputStage:
            { stage: 'IXSCAN',
              keyPattern: { borough: 1 },
              indexName: 'borough_1',
              isMultiKey: false,
              multiKeyPaths: { borough: [] },
              isUnique: false,
              isSparse: false,
              isPartial: false,
              indexVersion: 2,
              direction: 'forward',
              indexBounds: { borough: [ ['Bronx', 'Bronx'] ] } },
              rejectedPlans: [] },
          executionStats:
            { executionSuccess: true,
              nReturned: 309,
              executionTimeMillis: 0,
              totalKeysExamined: 309,
              totalDocsExamined: 309,
              executionStages:
                { stage: 'FETCH',
                  nReturned: 309,
                  executionTimeMillisEstimate: 0,
                  works: 310,
                  advanced: 309,
                  needTime: 0,
                  needYield: 0,
                  saveState: 0,
                  restoreState: 0,
                  isEOF: 1,
                  docsExamined: 309,
                  alreadyHasObj: 0,
                  inputStage:
                    { stage: 'IXSCAN',
                      nReturned: 309,
                      executionTimeMillisEstimate: 0,
                      works: 310,
                      advanced: 309,
                      needTime: 0,
```

```

needYield: 0,
saveState: 0,
restoreState: 0,
isEOF: 1,
keyPattern: { borough: 1 },
indexName: 'borough_1',
isMultiKey: false,
multiKeyPaths: { borough: [] },
isUnique: false,
isSparse: false,
isPartial: false,
indexVersion: 2,
direction: 'forward',
indexBounds: { borough: [ ['"Bronx"', '"Bronx"'] ] },
keysExamined: 309,
seeks: 1,
dupsTested: 0,
dupsDropped: 0 } } },
command:
{ find: 'restaurant',
  filter: { borough: 'Bronx' },
  '$db': 'hotel' },
serverInfo:
{ host: 'LAPTOP-SF40VR81',
  port: 27017,
  version: '5.0.12',
  gitVersion: '79cfcdd83eb6f64e164a588d0daf9bb873328b45' },
serverParameters:
{ internalQueryFacetBufferSizeBytes: 104857600,
  internalQueryFacetMaxOutputDocSizeBytes: 104857600,
  internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
  internalDocumentSourceGroupMaxMemoryBytes: 104857600,
  internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
  internalQueryProhibitBlockingMergeOnMongoS: 0,
  internalQueryMaxAddToSetBytes: 104857600,
  internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600 },
ok: 1 }

```

3) Compound index

```

> db.restaurant.createIndex({"name":1,"restaurant_id":1})
'name_1_restaurant_id_1'

```



```
> db.restaurant.find({"name":"African Market (Baboon
Cafe)","restaurant_id":40368026}).explain("queryPlanner")
{ explainVersion: '1',
  queryPlanner:
    { namespace: 'hotel.restaurant',
      indexFilterSet: false,
      parsedQuery:
        { '$and':
            [ [ { name: { '$eq': 'African Market (Baboon Cafe)' } },
              { restaurant_id: { '$eq': 40368026 } } ] ],
          queryHash: '6B8584F8',
          planCacheKey: 'E8E81FCF',
          maxIndexedOrSolutionsReached: false,
          maxIndexedAndSolutionsReached: false,
          maxScansToExplodeReached: false,
          winningPlan:
            { stage: 'FETCH',
              inputStage:
                { stage: 'IXSCAN',
                  keyPattern: { name: 1, restaurant_id: 1 },
                  indexName: 'name_1_restaurant_id_1',
                  isMultiKey: false,
                  multiKeyPaths: { name: [], restaurant_id: [] },
                  isUnique: false,
                  isSparse: false,
                  isPartial: false,
                  indexVersion: 2,
                  direction: 'forward',
                  indexBounds:
                    { name: [ ['African Market (Baboon Cafe)', 'African Market (Baboon Cafe)'] ],
                      restaurant_id: [ '[40368026, 40368026]' ] } },
                  rejectedPlans: [] },
              command:
                { find: 'restaurant',
                  filter:
                    { name: 'African Market (Baboon Cafe)',
                      restaurant_id: 40368026 },
                  '$db': 'hotel' },
              serverInfo:
                { host: 'LAPTOP-SF40VR81',
                  port: 27017,
                  version: '5.0.12',
                  gitVersion: '79cfcdd83eb6f64e164a588d0daf9bb873328b45' },
              serverParameters:
                { internalQueryFacetBufferSizeBytes: 104857600,
                  internalQueryFacetMaxOutputDocSizeBytes: 104857600,
                  internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
                  internalDocumentSourceGroupMaxMemoryBytes: 104857600,
                  internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
```

```
internalQueryProhibitBlockingMergeOnMongoS: 0,  
internalQueryMaxAddToSetBytes: 104857600,  
internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600 },  
ok: 1 }
```

```
> db.restaurant.find({"name":"African Market (Baboon  
Cafe)","restaurant_id":40368026}).explain("executionStats")  
{ explainVersion: '1',  
  queryPlanner:  
    { namespace: 'hotel.restaurant',  
      indexFilterSet: false,  
      parsedQuery:  
        { '$and':  
          [ { name: { '$eq': 'African Market (Baboon Cafe)' } },  
            { restaurant_id: { '$eq': 40368026 } } ] },  
          maxIndexedOrSolutionsReached: false,  
          maxIndexedAndSolutionsReached: false,  
          maxScansToExplodeReached: false,  
          winningPlan:  
            { stage: 'FETCH',  
              inputStage:  
                { stage: 'IXSCAN',  
                  keyPattern: { name: 1, restaurant_id: 1 },  
                  indexName: 'name_1_restaurant_id_1',  
                  isMultiKey: false,  
                  multiKeyPaths: { name: [], restaurant_id: [] },  
                  isUnique: false,  
                  isSparse: false,  
                  isPartial: false,  
                  indexVersion: 2,  
                  direction: 'forward',  
                  indexBounds:  
                    { name: [ '['African Market (Baboon Cafe)', 'African Market (Baboon Cafe)'] ],  
                      restaurant_id: [ '[40368026, 40368026]' ] } } },  
                  rejectedPlans: [] },  
          executionStats:  
            { executionSuccess: true,  
              nReturned: 0,  
              executionTimeMillis: 4,  
              totalKeysExamined: 0,  
              totalDocsExamined: 0,  
              executionStages:  
                { stage: 'FETCH',  
                  nReturned: 0,  
                  executionTimeMillisEstimate: 0,  
                  works: 1,  
                  advanced: 0,  
                  needTime: 0,  
                  needYield: 0,
```

```
saveState: 0,
restoreState: 0,
isEOF: 1,
docsExamined: 0,
alreadyHasObj: 0,
inputStage:
{ stage: 'IXSCAN',
nReturned: 0,
executionTimeMillisEstimate: 0,
works: 1,
advanced: 0,
needTime: 0,
needYield: 0,
saveState: 0,
restoreState: 0,
isEOF: 1,
keyPattern: { name: 1, restaurant_id: 1 },
indexName: 'name_1_restaurant_id_1',
isMultiKey: false,
multiKeyPaths: { name: [], restaurant_id: [] },
isUnique: false,
isSparse: false,
isPartial: false,
indexVersion: 2,
direction: 'forward',
indexBounds:
{ name: [ '['African Market (Baboon Cafe)', 'African Market (Baboon Cafe)'] ],
restaurant_id: [ '[40368026, 40368026]' ] },
keysExamined: 0,
seeks: 1,
dupsTested: 0,
dupsDropped: 0 } } },
command:
{ find: 'restaurant',
filter:
{ name: 'African Market (Baboon Cafe)',
restaurant_id: 40368026 },
'$db': 'hotel' },
serverInfo:
{ host: 'LAPTOP-SF40VR81',
port: 27017,
version: '5.0.12',
gitVersion: '79cfdd83eb6f64e164a588d0daf9bb873328b45' },
serverParameters:
{ internalQueryFacetBufferSizeBytes: 104857600,
internalQueryFacetMaxOutputDocSizeBytes: 104857600,
internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
internalDocumentSourceGroupMaxMemoryBytes: 104857600,
internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
internalQueryProhibitBlockingMergeOnMongoS: 0,
```

```
internalQueryMaxAddToSetBytes: 104857600,  
internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600 },  
ok: 1 }
```

```
>db.restaurant.find({"name":"African Market (Baboon  
Cafe)","restaurant_id":40368026}).explain("allPlansExecution")  
{ explainVersion: '1',  
  queryPlanner:  
    { namespace: 'hotel.restaurant',  
      indexFilterSet: false,  
      parsedQuery:  
        { '$and':  
          [ { name: { '$eq': 'African Market (Baboon Cafe)' } },  
            { restaurant_id: { '$eq': 40368026 } } ] },  
          maxIndexedOrSolutionsReached: false,  
          maxIndexedAndSolutionsReached: false,  
          maxScansToExplodeReached: false,  
          winningPlan:  
            { stage: 'FETCH',  
              inputStage:  
                { stage: 'IXSCAN',  
                  keyPattern: { name: 1, restaurant_id: 1 },  
                  indexName: 'name_1_restaurant_id_1',  
                  isMultiKey: false,  
                  multiKeyPaths: { name: [], restaurant_id: [] },  
                  isUnique: false,  
                  isSparse: false,  
                  isPartial: false,  
                  indexVersion: 2,  
                  direction: 'forward',  
                  indexBounds:  
                    { name: [ '['African Market (Baboon Cafe)', 'African Market (Baboon Cafe)'] ],  
                      restaurant_id: [ '[40368026, 40368026]' ] } } },  
                  rejectedPlans: [] },  
          executionStats:  
            { executionSuccess: true,  
              nReturned: 0,  
              executionTimeMillis: 0,  
              totalKeysExamined: 0,  
              totalDocsExamined: 0,  
              executionStages:  
                { stage: 'FETCH',  
                  nReturned: 0,  
                  executionTimeMillisEstimate: 0,  
                  works: 1,  
                  advanced: 0,  
                  needTime: 0,  
                  needYield: 0,  
                  saveState: 0,
```

```
restoreState: 0,
isEOF: 1,
docsExamined: 0,
alreadyHasObj: 0,
inputStage:
{ stage: 'IXSCAN',
nReturned: 0,
executionTimeMillisEstimate: 0,
works: 1,
advanced: 0,
needTime: 0,
needYield: 0,
saveState: 0,
restoreState: 0,
isEOF: 1,
keyPattern: { name: 1, restaurant_id: 1 },
indexName: 'name_1_restaurant_id_1',
isMultiKey: false,
multiKeyPaths: { name: [], restaurant_id: [] },
isUnique: false,
isSparse: false,
isPartial: false,
indexVersion: 2,
direction: 'forward',
indexBounds:
{ name: [ '['African Market (Baboon Cafe)', 'African Market (Baboon Cafe)']' ],
restaurant_id: [ '[40368026, 40368026]' ] },
keysExamined: 0,
seeks: 1,
dupsTested: 0,
dupsDropped: 0 } },
allPlansExecution: [ ] },
command:
{ find: 'restaurant',
filter:
{ name: 'African Market (Baboon Cafe)',
restaurant_id: 40368026 },
'$db': 'hotel' },
serverInfo:
{ host: 'LAPTOP-SF40VR81',
port: 27017,
version: '5.0.12',
gitVersion: '79cfcd83eb6f64e164a588d0daf9bb873328b45' },
serverParameters:
{ internalQueryFacetBufferSizeBytes: 104857600,
internalQueryFacetMaxOutputDocSizeBytes: 104857600,
internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
internalDocumentSourceGroupMaxMemoryBytes: 104857600,
internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
internalQueryProhibitBlockingMergeOnMongoS: 0,
```

```
internalQueryMaxAddToSetBytes: 104857600,  
internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600 },  
ok: 1 }
```

3. Multikey index

```
> db.restaurant.createIndex({address:1})  
'address_1'
```

```
> db.restaurant.find({"building":284,"coord":[-73.9829239, 40.6580753 ],  
"street": "Prospect Park West","zipcode": 11215}).explain("queryPlanner")  
{ explainVersion: '1',  
  queryPlanner:  
    { namespace: 'hotel.restaurant',  
      indexFilterSet: false,  
      parsedQuery:  
        { '$and':  
          [ { building: { '$eq': 284 } },  
            { coord: { '$eq': [-73.9829239, 40.6580753 ] } },  
            { street: { '$eq': 'Prospect Park West' } },  
            { zipcode: { '$eq': 11215 } } ] },  
          queryHash: '7063889A',  
          planCacheKey: 'EE06BCFA',  
          maxIndexedOrSolutionsReached: false,  
          maxIndexedAndSolutionsReached: false,  
          maxScansToExplodeReached: false,  
          winningPlan:  
            { stage: 'COLLSCAN',  
              filter:  
                { '$and':  
                  [ { building: { '$eq': 284 } },  
                    { coord: { '$eq': [-73.9829239, 40.6580753 ] } },  
                    { street: { '$eq': 'Prospect Park West' } },  
                    { zipcode: { '$eq': 11215 } } ] },  
                  direction: 'forward' },  
                rejectedPlans: [] },  
              command:  
                { find: 'restaurant',  
                  filter:  
                    { building: 284,  
                      coord: [-73.9829239, 40.6580753 ],  
                      street: 'Prospect Park West',  
                      zipcode: 11215 },  
                  '$db': 'hotel' },  
                  serverInfo:
```

```
{ host: 'LAPTOP-SF40VR81',
port: 27017,
version: '5.0.12',
gitVersion: '79cfcdd83eb6f64e164a588d0daf9bb873328b45' },
serverParameters:
{ internalQueryFacetBufferSizeBytes: 104857600,
internalQueryFacetMaxOutputDocSizeBytes: 104857600,
internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
internalDocumentSourceGroupMaxMemoryBytes: 104857600,
internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
internalQueryProhibitBlockingMergeOnMongoS: 0,
internalQueryMaxAddToSetBytes: 104857600,
internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600 },
ok: 1 }
```

```
> db.restaurant.find({"building":284,"coord":[-73.9829239, 40.6580753 ],
"street": "Prospect Park West","zipcode": 11215}).explain("executionStats")
{ explainVersion: '1',
queryPlanner:
{ namespace: 'hotel.restaurant',
indexFilterSet: false,
parsedQuery:
{ '$and':
[ { building: { '$eq': 284 } },
{ coord: { '$eq': [-73.9829239, 40.6580753 ] } },
{ street: { '$eq': 'Prospect Park West' } },
{ zipcode: { '$eq': 11215 } } ] },
maxIndexedOrSolutionsReached: false,
maxIndexedAndSolutionsReached: false,
maxScansToExplodeReached: false,
winningPlan:
{ stage: 'COLLSCAN',
filter:
{ '$and':
[ { building: { '$eq': 284 } },
{ coord: { '$eq': [-73.9829239, 40.6580753 ] } },
{ street: { '$eq': 'Prospect Park West' } },
{ zipcode: { '$eq': 11215 } } ] },
direction: 'forward' },
rejectedPlans: [] },
executionStats:
{ executionSuccess: true,
nReturned: 0,
executionTimeMillis: 1,
totalKeysExamined: 0,
totalDocsExamined: 3772,
executionStages:
{ stage: 'COLLSCAN',
filter:
```

```

{ '$and':
[ { building: { '$eq': 284 } },
  { coord: { '$eq': [ -73.9829239, 40.6580753 ] } },
  { street: { '$eq': 'Prospect Park West' } },
  { zipcode: { '$eq': 11215 } } ] },
nReturned: 0,
executionTimeMillisEstimate: 0,
works: 3774,
advanced: 0,
needTime: 3773,
needYield: 0,
saveState: 3,
restoreState: 3,
isEOF: 1,
direction: 'forward',
docsExamined: 3772 } },
command:
{ find: 'restaurant',
  filter:
  { building: 284,
    coord: [ -73.9829239, 40.6580753 ],
    street: 'Prospect Park West',
    zipcode: 11215 },
    '$db': 'hotel' },
  serverInfo:
  { host: 'LAPTOP-SF40VR81',
    port: 27017,
    version: '5.0.12',
    gitVersion: '79cfcdd83eb6f64e164a588d0daf9bb873328b45' },
    serverParameters:
    { internalQueryFacetBufferSizeBytes: 104857600,
      internalQueryFacetMaxOutputDocSizeBytes: 104857600,
      internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
      internalDocumentSourceGroupMaxMemoryBytes: 104857600,
      internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
      internalQueryProhibitBlockingMergeOnMongoS: 0,
      internalQueryMaxAddToSetBytes: 104857600,
      internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600 },
      ok: 1 }

```

```

> db.restaurant.find({"building":284,"coord":[ -73.9829239, 40.6580753
], "street": "Prospect Park West","zipcode":
11215}).explain("allPlansExecution")
{ explainVersion: '1',
  queryPlanner:
  { namespace: 'hotel.restaurant',
    indexFilterSet: false,
    parsedQuery:
    { '$and':

```



```
[ { building: { '$eq': 284 } },
  { coord: { '$eq': [ -73.9829239, 40.6580753 ] } },
  { street: { '$eq': 'Prospect Park West' } },
  { zipcode: { '$eq': 11215 } } ] },
maxIndexedOrSolutionsReached: false,
maxIndexedAndSolutionsReached: false,
maxScansToExplodeReached: false,
winningPlan:
{ stage: 'COLLSCAN',
  filter:
  { '$and':
  [ { building: { '$eq': 284 } },
    { coord: { '$eq': [ -73.9829239, 40.6580753 ] } },
    { street: { '$eq': 'Prospect Park West' } },
    { zipcode: { '$eq': 11215 } } ] },
    direction: 'forward' },
  rejectedPlans: [] },
  executionStats:
  { executionSuccess: true,
    nReturned: 0,
    executionTimeMillis: 3,
    totalKeysExamined: 0,
    totalDocsExamined: 3772,
    executionStages:
    { stage: 'COLLSCAN',
      filter:
      { '$and':
      [ { building: { '$eq': 284 } },
        { coord: { '$eq': [ -73.9829239, 40.6580753 ] } },
        { street: { '$eq': 'Prospect Park West' } },
        { zipcode: { '$eq': 11215 } } ] },
        nReturned: 0,
        executionTimeMillisEstimate: 0,
        works: 3774,
        advanced: 0,
        needTime: 3773,
        needYield: 0,
        saveState: 3,
        restoreState: 3,
        isEOF: 1,
        direction: 'forward',
        docsExamined: 3772 },
        allPlansExecution: [] },
        command:
        { find: 'restaurant',
          filter:
          { building: 284,
            coord: [ -73.9829239, 40.6580753 ],
            street: 'Prospect Park West',
            zipcode: 11215 },
```

```

'$db': 'hotel' },
serverInfo:
{ host: 'LAPTOP-SF40VR81',
port: 27017,
version: '5.0.12',
gitVersion: '79cfcdd83eb6f64e164a588d0daf9bb873328b45' },
serverParameters:
{ internalQueryFacetBufferSizeBytes: 104857600,
internalQueryFacetMaxOutputDocSizeBytes: 104857600,
internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
internalDocumentSourceGroupMaxMemoryBytes: 104857600,
internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
internalQueryProhibitBlockingMergeOnMongoS: 0,
internalQueryMaxAddToSetBytes: 104857600,
internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600 },
ok: 1 }

```

4. Show existing indexes

```

> db.restaurant.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_ ' },
  { v: 2, key: { borough: 1 }, name: 'borough_1' },
  {
    v: 2,
    key: { name: 1, restaurant_id: 1 },
    name: 'name_1_restaurant_id_1'
  },
  { v: 2, key: { address: 1 }, name: 'address_1' }
]

```

Special query:

```
db.Hotel.updateMany({ "Cuisines": "CHinese" }, { "$set": { "Cuisines.$": "Chinese" } })
```