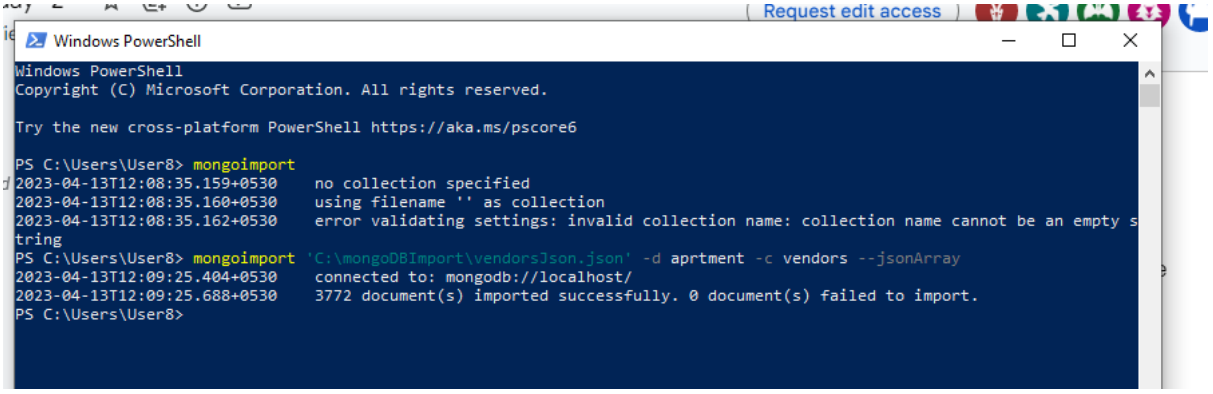
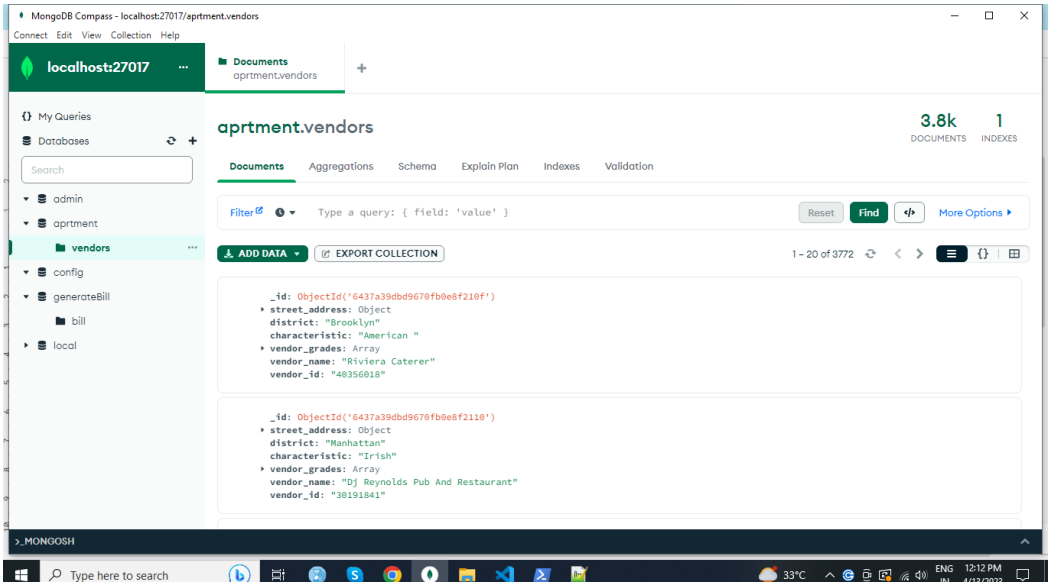
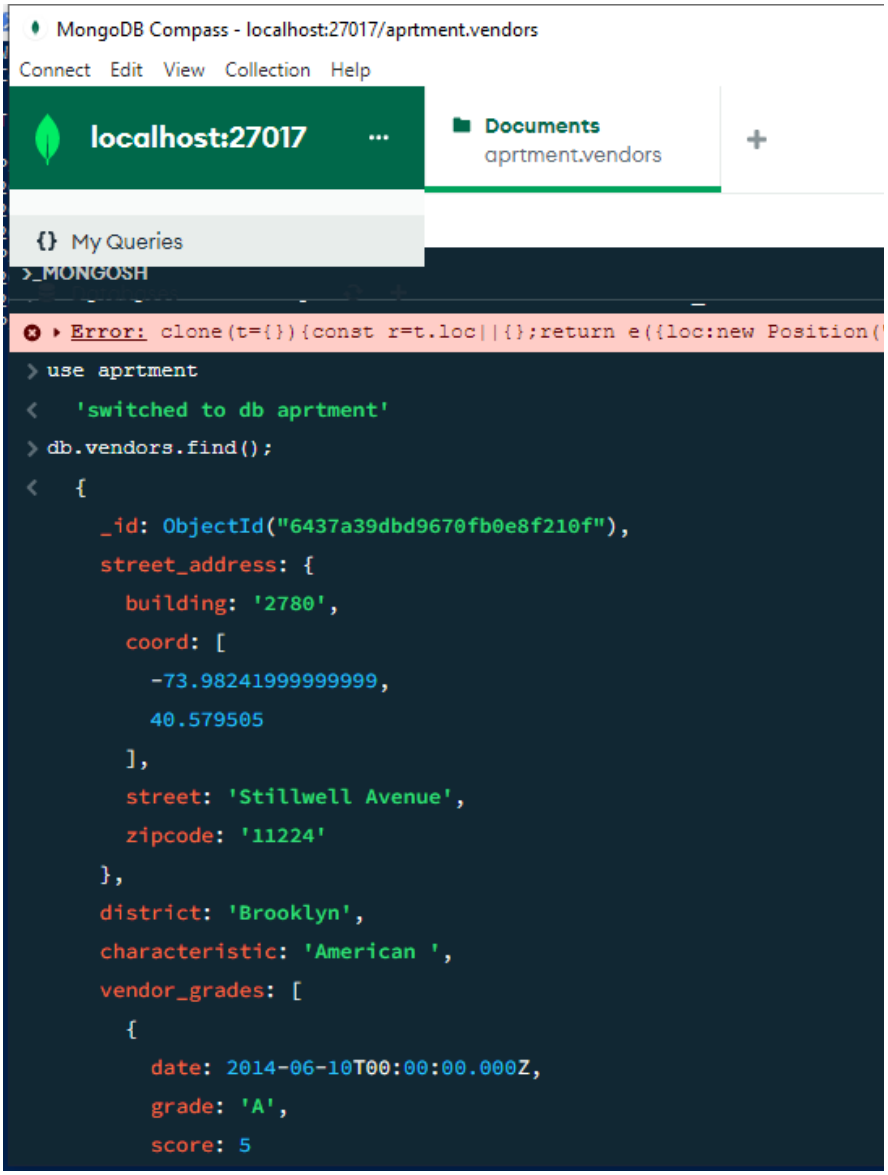
1. Import vendorsJson file in database with vendors collection



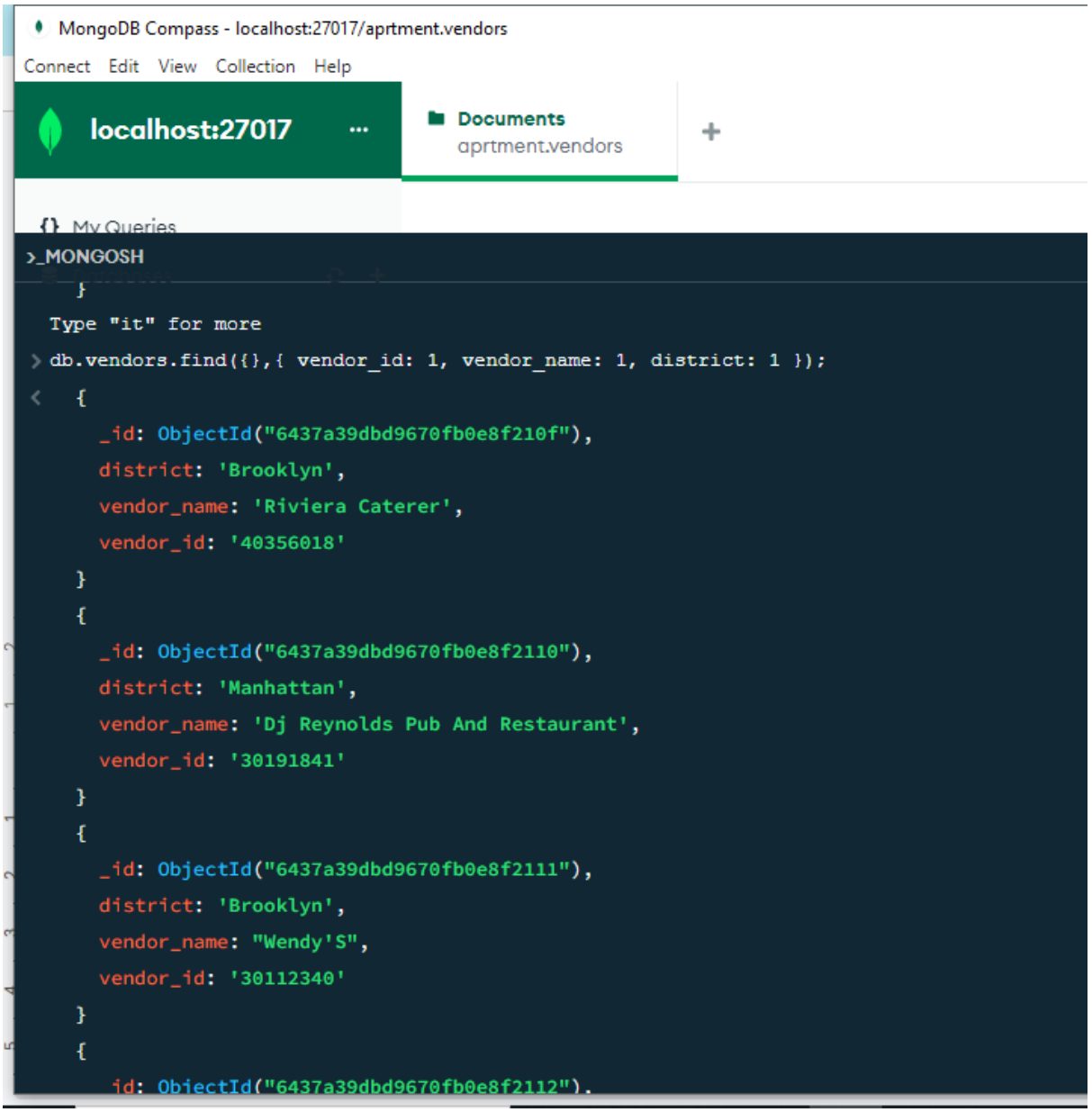


2.Display **all** the documents.



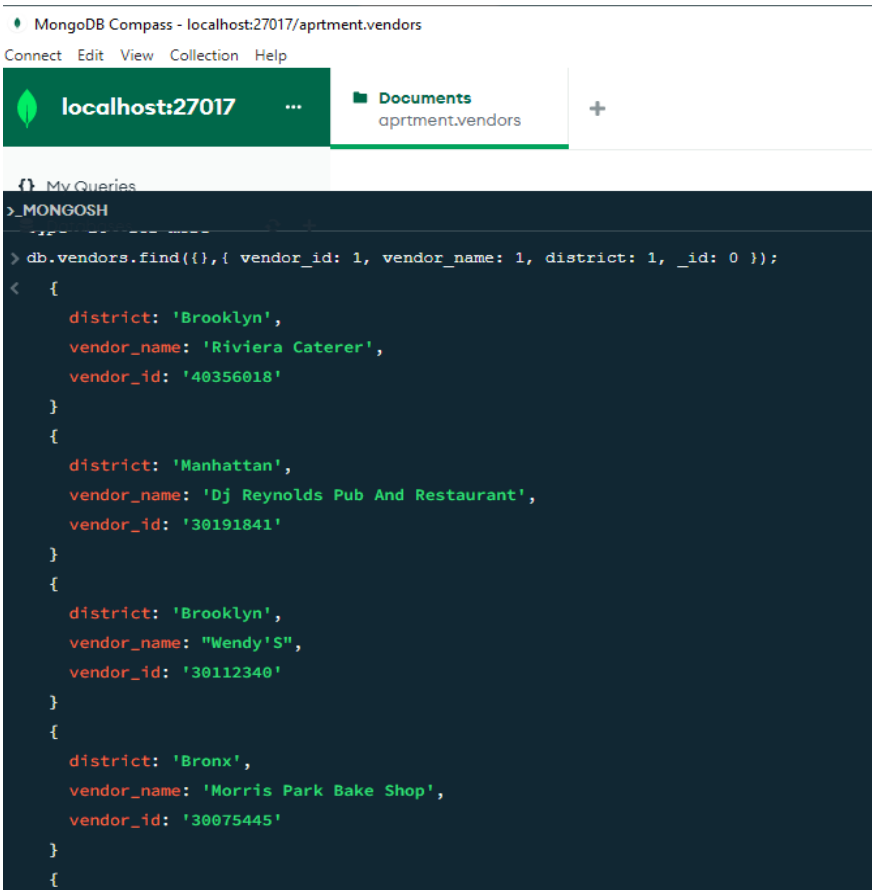
3. display the **fields vendor_id, vendor_name, district** for all the documents in the collection.

```
db.vendors.find({},{ vendor_id: 1, vendor_name: 1, district: 1 });
```



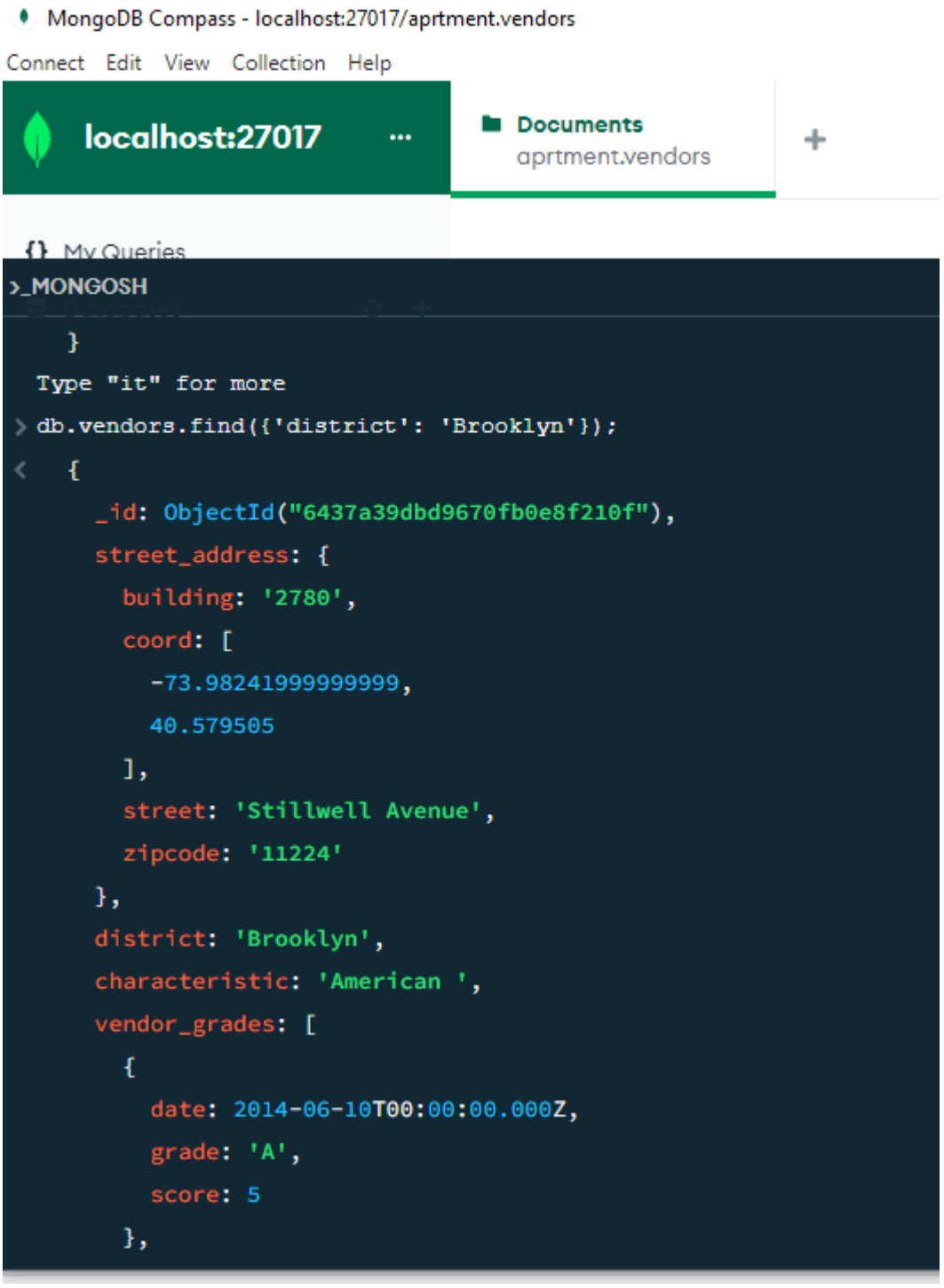
4.display the fields **vendor_id, vendor_name,street_address.building** but remove the field **_id** for all the documents in output.

```
db.vendors.find({},{ vendor_id: 1, vendor_name: 1, district: 1, _id: 0 });
```



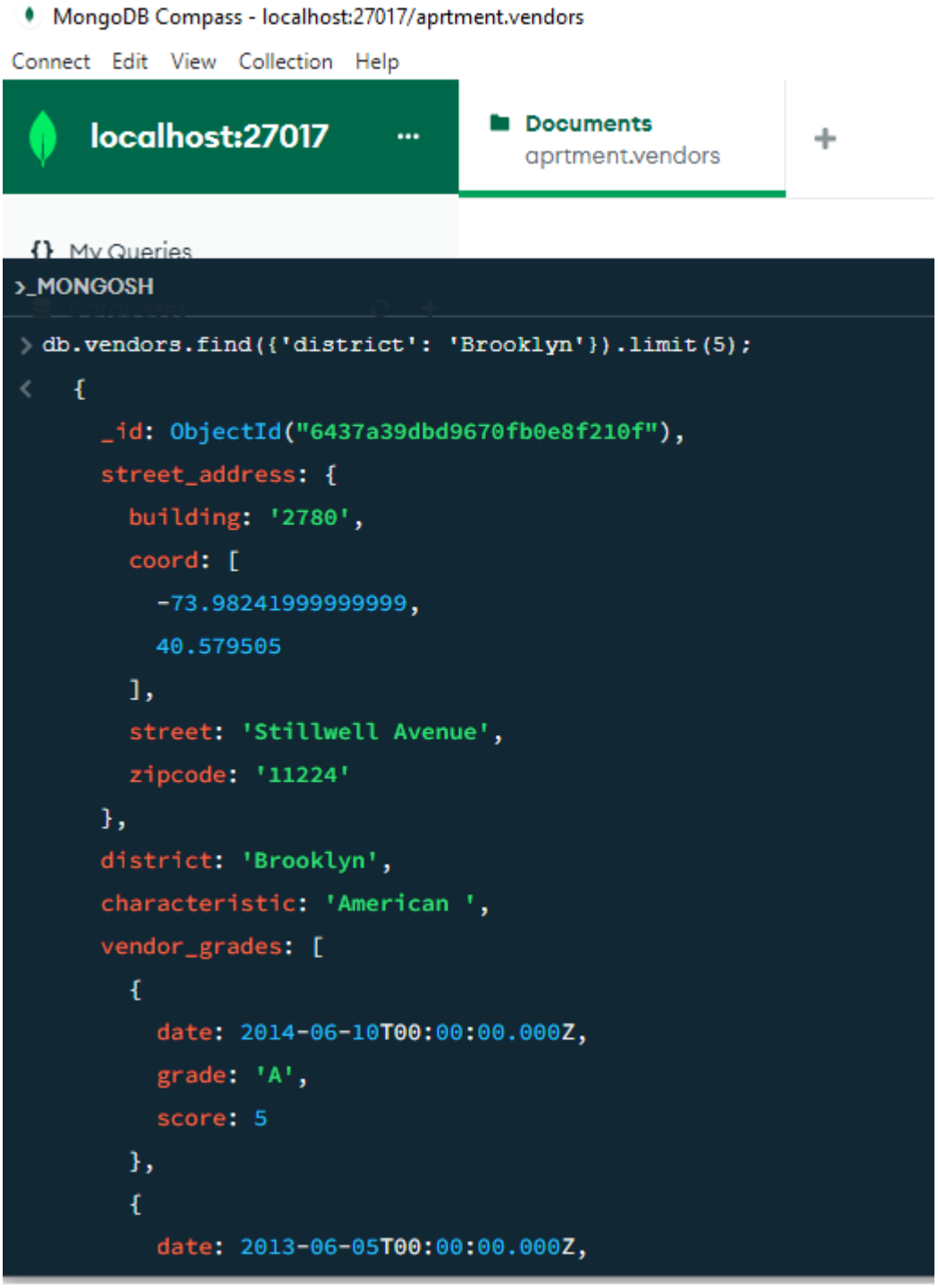
5. display all the vendors which is in the **district 'Brooklyn'**

```
db.vendors.find({'district': 'Brooklyn'});
```



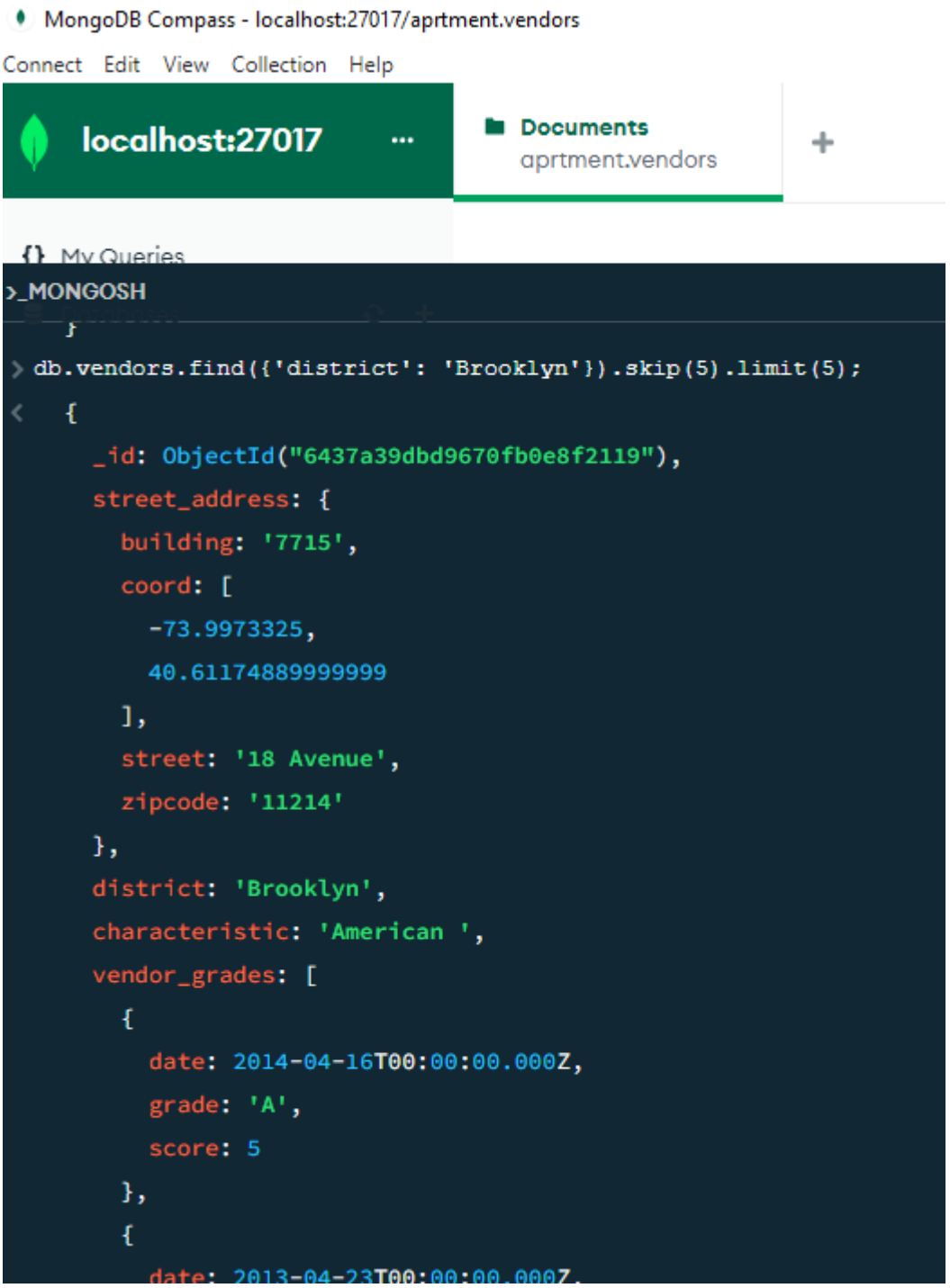
6. display the **first 5 vendors** which is in the **district ‘Brooklyn’**

```
db.vendors.find({'district': 'Brooklyn'}).limit(5);
```



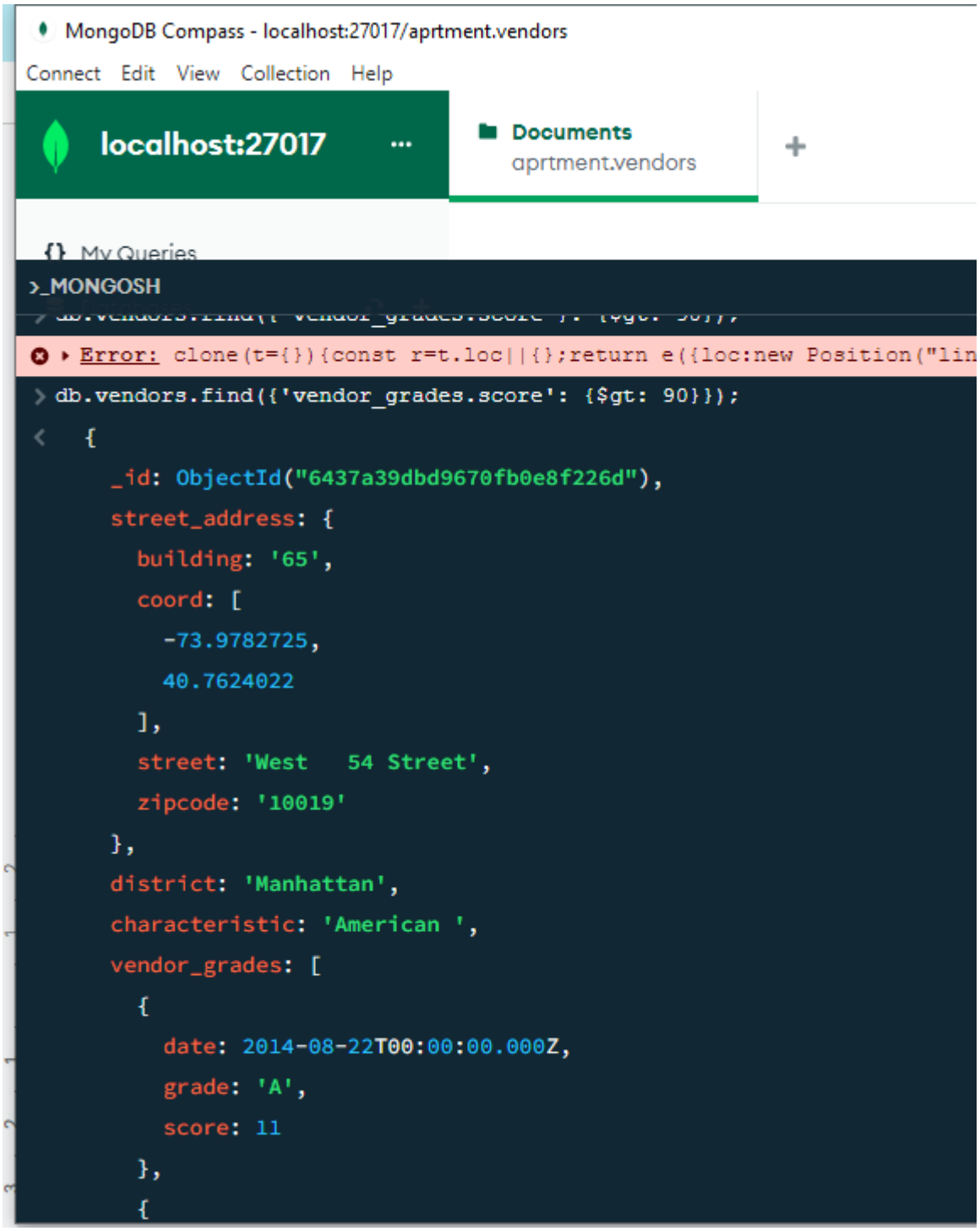
7. display the **next 5 vendors** after **skipping first 5** which is in the district ‘Brooklyn’

```
db.vendors.find({'district': 'Brooklyn'}).skip(5).limit(5);
```



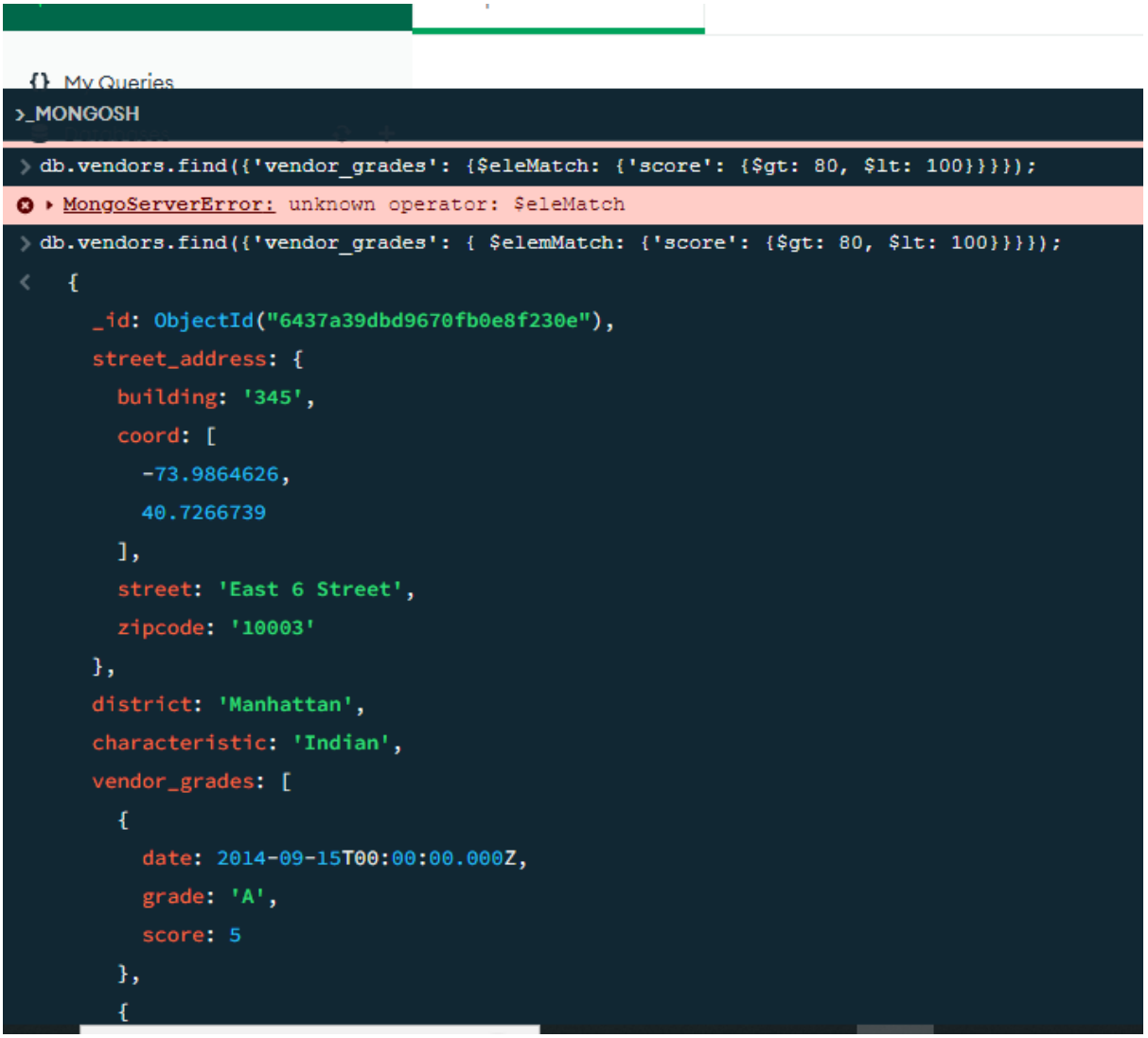
8. find the vendors who achieved a score **more than 90 in vendor_grades**

```
db.vendors.find({'vendor_grades.score': {$gt: 90}});
```



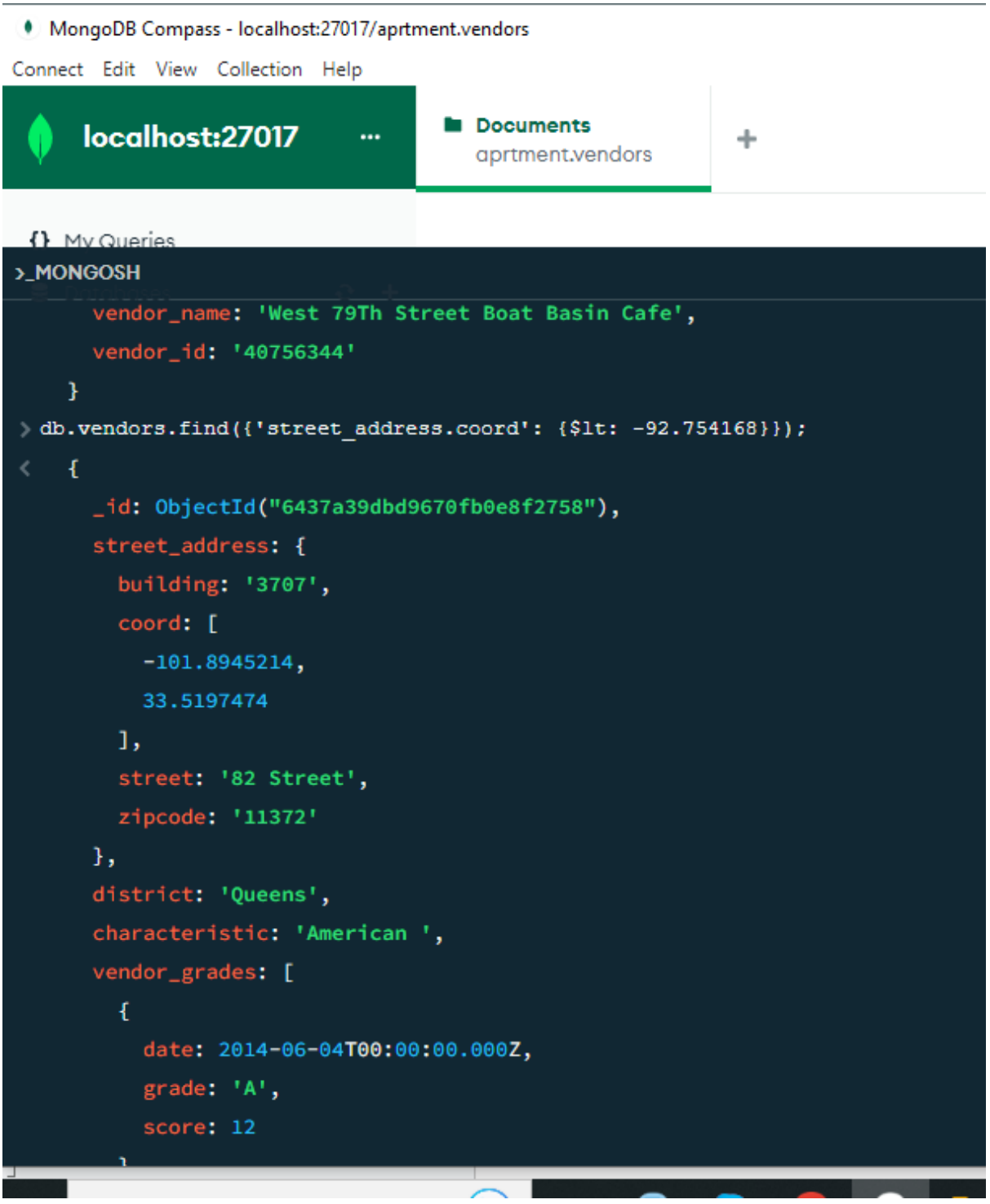
9. find the vendors that achieved a score, **more than 80 but less than 100.**

```
db.vendors.find({'vendor_grades': { $elemMatch: {'score': {$gt: 80, $lt: 100}}}});
```



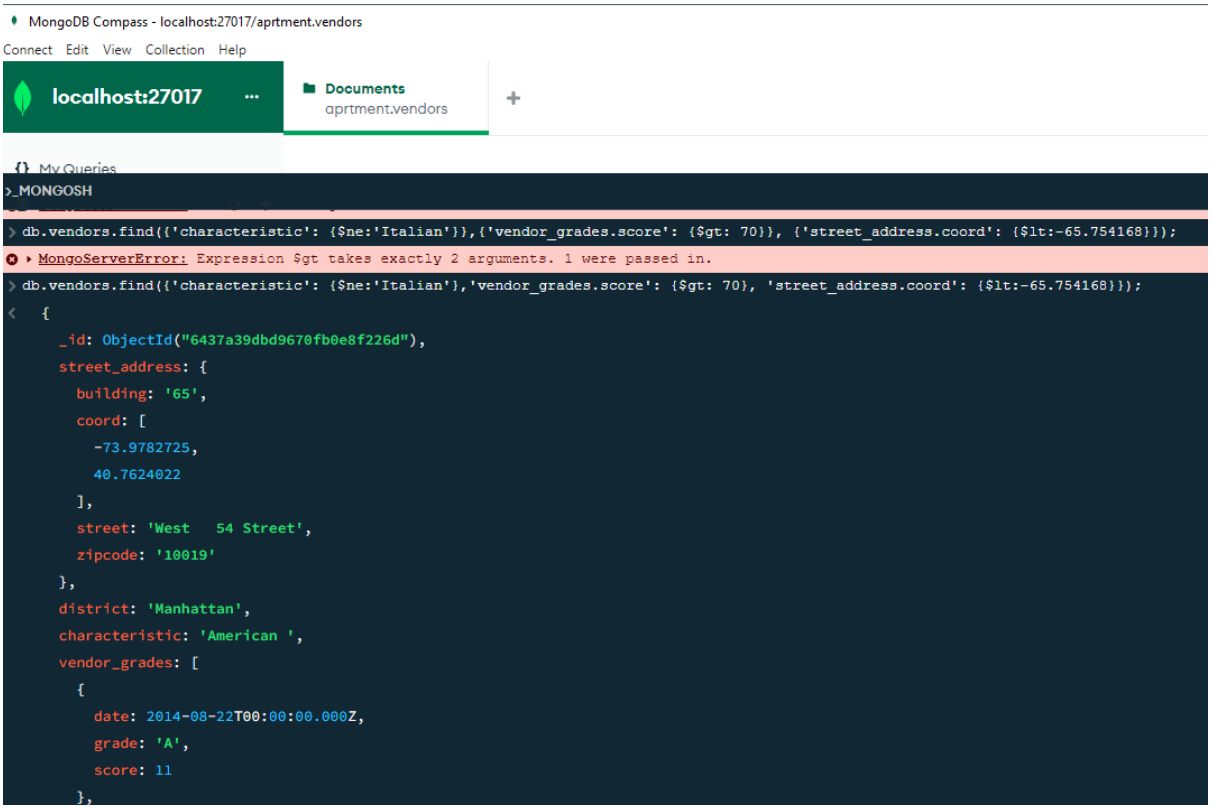
10. find the vendors which are located in **latitude(coord) value less than -92.754168**.

```
db.vendors.find({'street_address.coord': {$lt: -92.754168}});
```



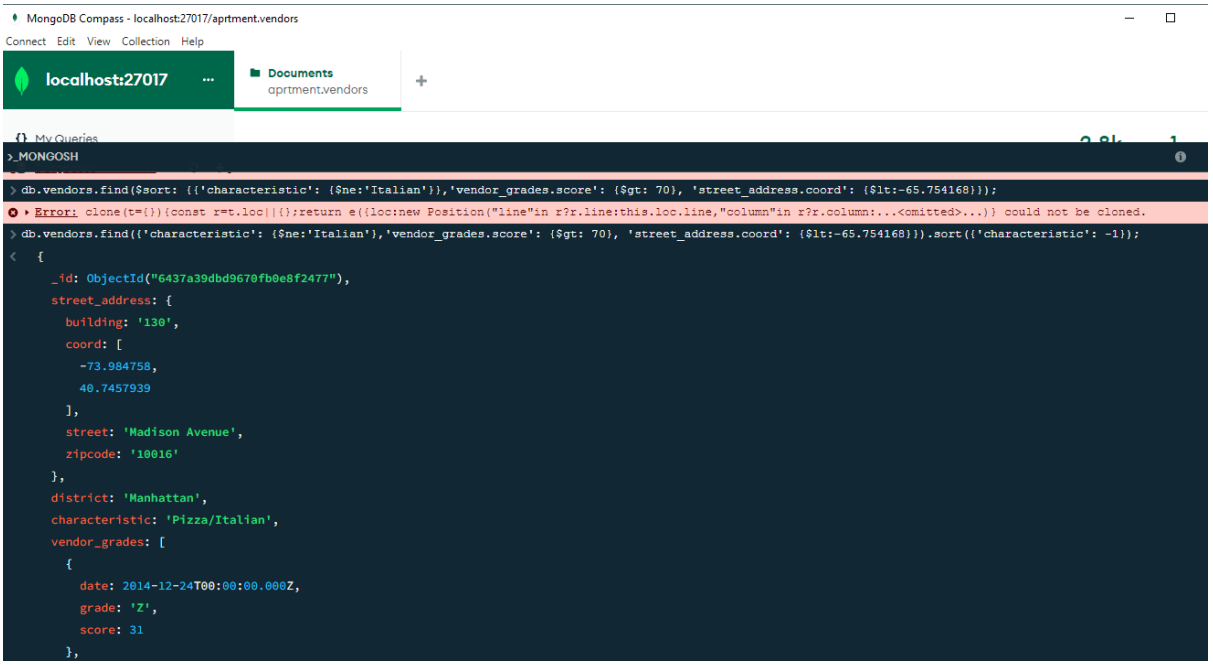
11. find the vendors that characteristic is not ‘Italian' and their grade score **more than 70 and latitude less than -65.754168**

```
db.vendors.find({'characteristic': {$ne:'Italian'},'vendor_grades.score': {$gt: 70}, 'street_address.coord': {$lt:-65.754168}});
```



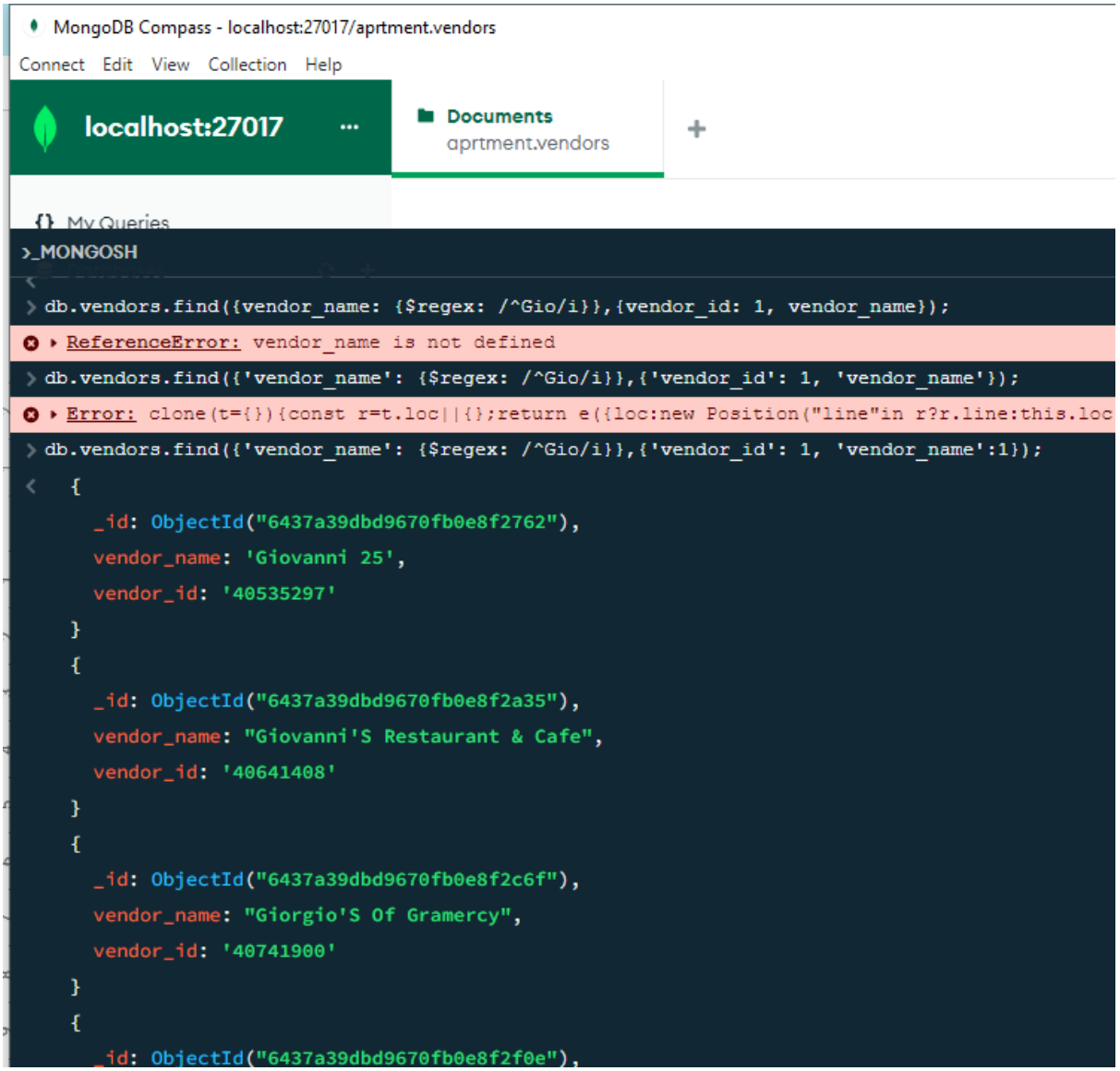
12. find the vendors that characteristic is not ‘Italian' and their grade score more than 70 and latitude less than -65.754168.The document must be displayed according to the characteristic in descending order.

```
db.vendors.find({'characteristic': {$ne:'Italian'},'vendor_grades.score': {$gt: 70}, 'street_address.coord': {$lt:-65.754168}}).sort({'characteristic': -1});
```



13. find the vendor_id, vendor_name, for those vendors which contain 'Gio' as first three letters for its name.

```
db.vendors.find({'vendor_name': {$regex: /^Gio/i}},{'vendor_id': 1, 'vendor_name':1});
```

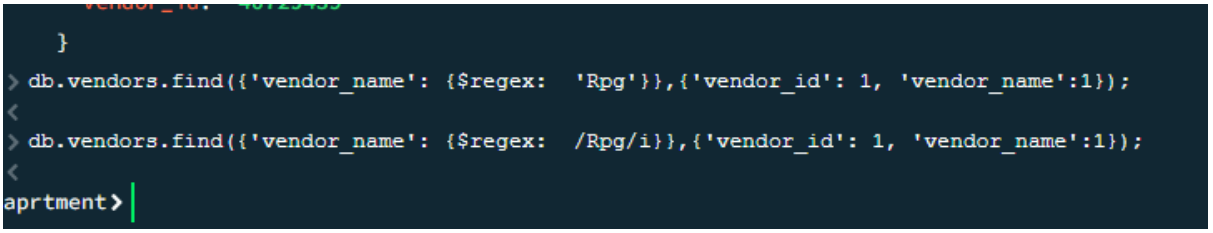


14. find the vendor_id, vendor_name, for those vendors which contain 'cas' as last three letters for its name.

```
db.vendors.find({'vendor_name': {$regex:  'cas$'}},{'vendor_id': 1, 'vendor_name':1});
```

15. find the vendor_id, vendor_name, for those vendors which contain 'Rpg' as three letters somewhere in its name

```
db.vendors.find({'vendor_name': {$regex:  'Rpg'}},{'vendor_id': 1, 'vendor_name':1});
```



16. arrange the name of the vendors in ascending order along with all the columns.

```
db.vendors.find().sort({'vendor_name':1});
```

17. arrange the name of the vendors in descending order along with all the columns.

```
db.vendors.find().sort({'vendor_name':-1});
```

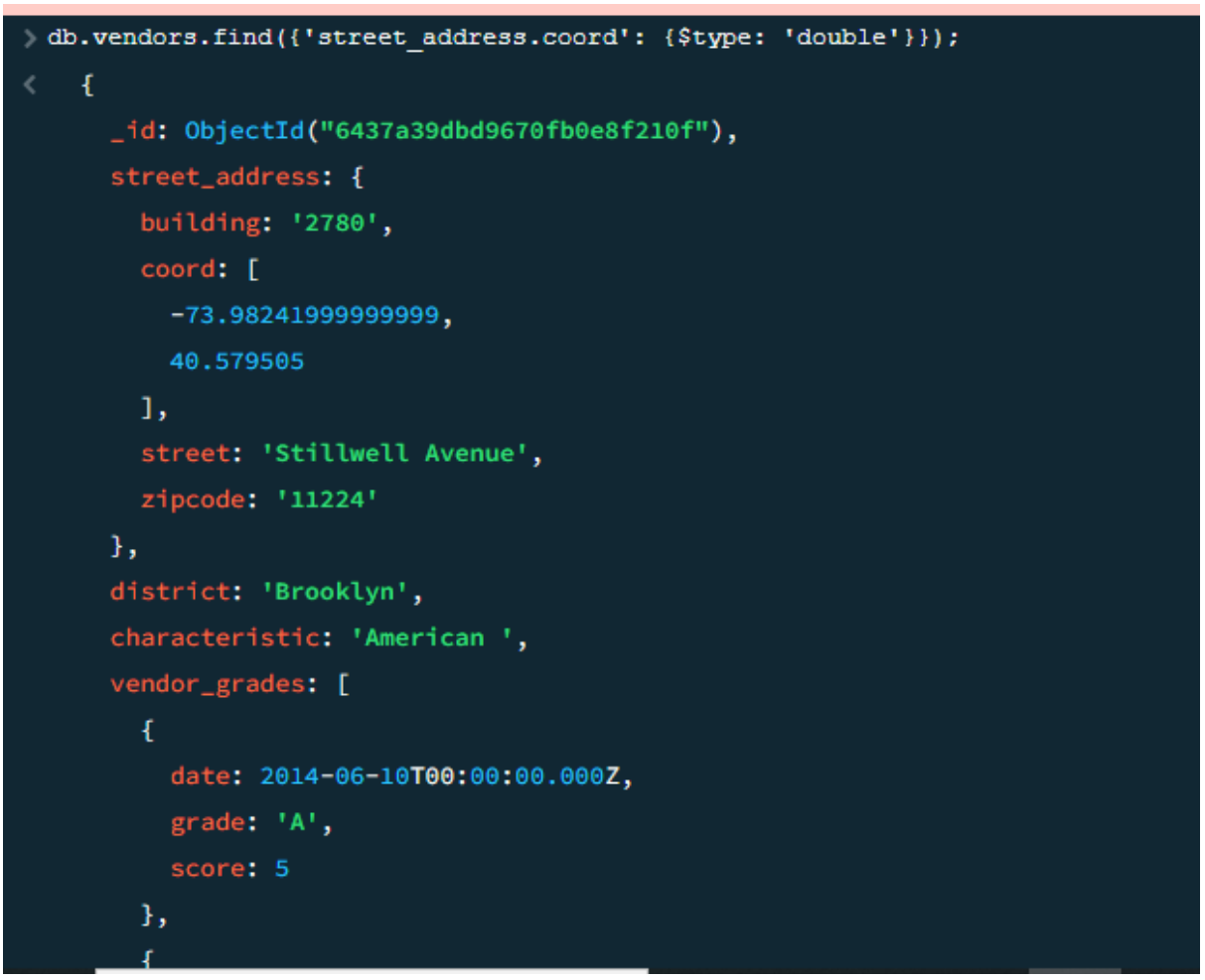
18. query to know whether all the street_address contains the street or not.

```
db.vendors.find({'street_address.street': {$exists : true}});
```

.

19. query which will select all documents in the vendors collection where the ‘coord’ field value is Double.

```
db.vendors.find({'street_address.coord': {$type: 'double'}});
```



20. find the vendors that have at least one grade with a score of less than 5 and that are located in the district of Manhattan or Brooklyn, and their characteristics are not Italian.

```
db.vendors.find({$and: [{ "vendor_grades.score": { $lt: 5 } }, { "characteristic": { $ne: "Italian" } },{ $or: [{ "district": "Manhattan" }, { "district": "Brooklyn" }] }]});
```