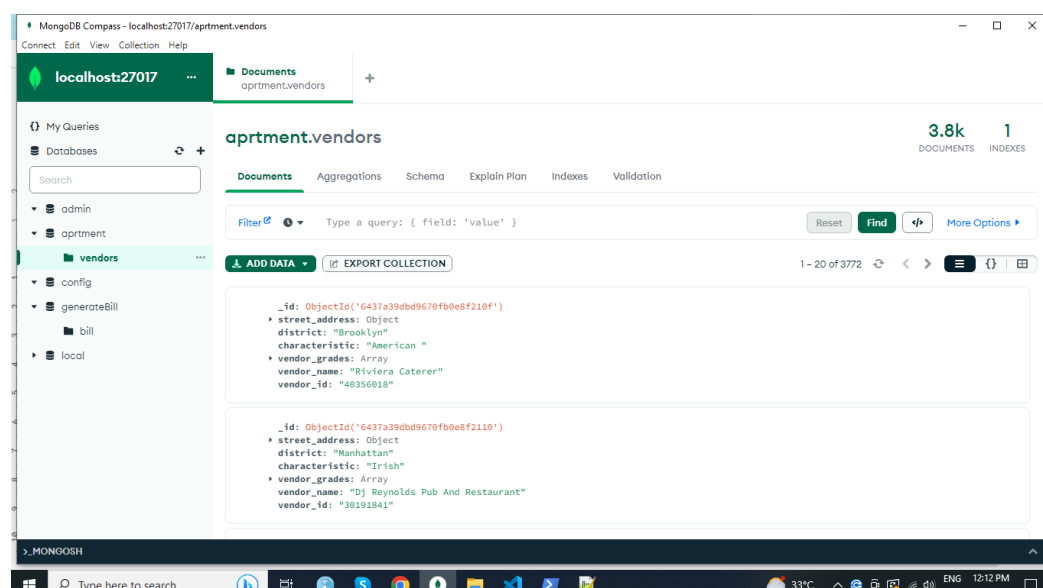


## 1. Import vendorsJson file in database with vendors collection

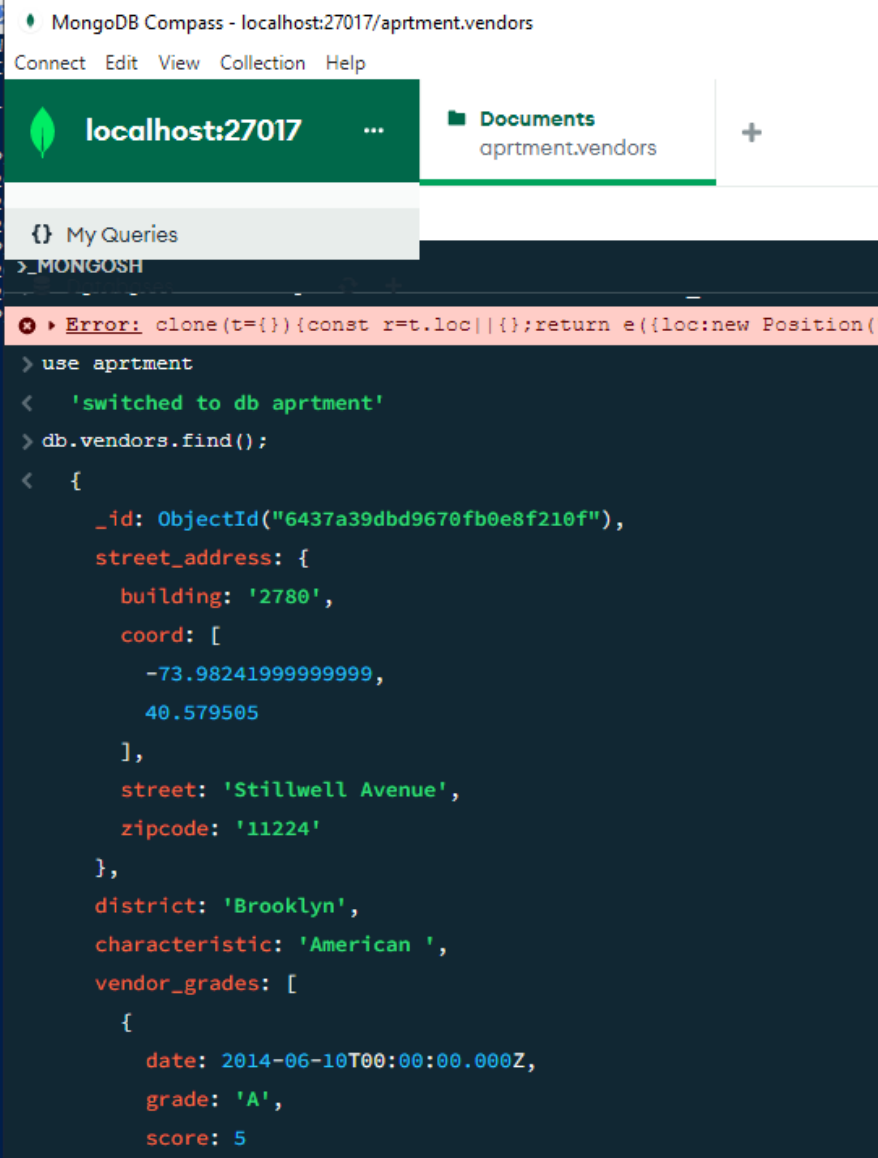
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\User8> mongoimport
2023-04-13T12:08:35.159+0530 no collection specified
2023-04-13T12:08:35.160+0530 using filename '' as collection
2023-04-13T12:08:35.162+0530 error validating settings: invalid collection name: collection name cannot be an empty string
PS C:\Users\User8> mongoimport 'C:\mongoDBImport\vendorsJson.json' -d aptment -c vendors --jsonArray
2023-04-13T12:09:25.404+0530 connected to: mongodb://localhost/
2023-04-13T12:09:25.688+0530 3772 document(s) imported successfully. 0 document(s) failed to import.
PS C:\Users\User8>
```



## 2.Display **all** the documents.

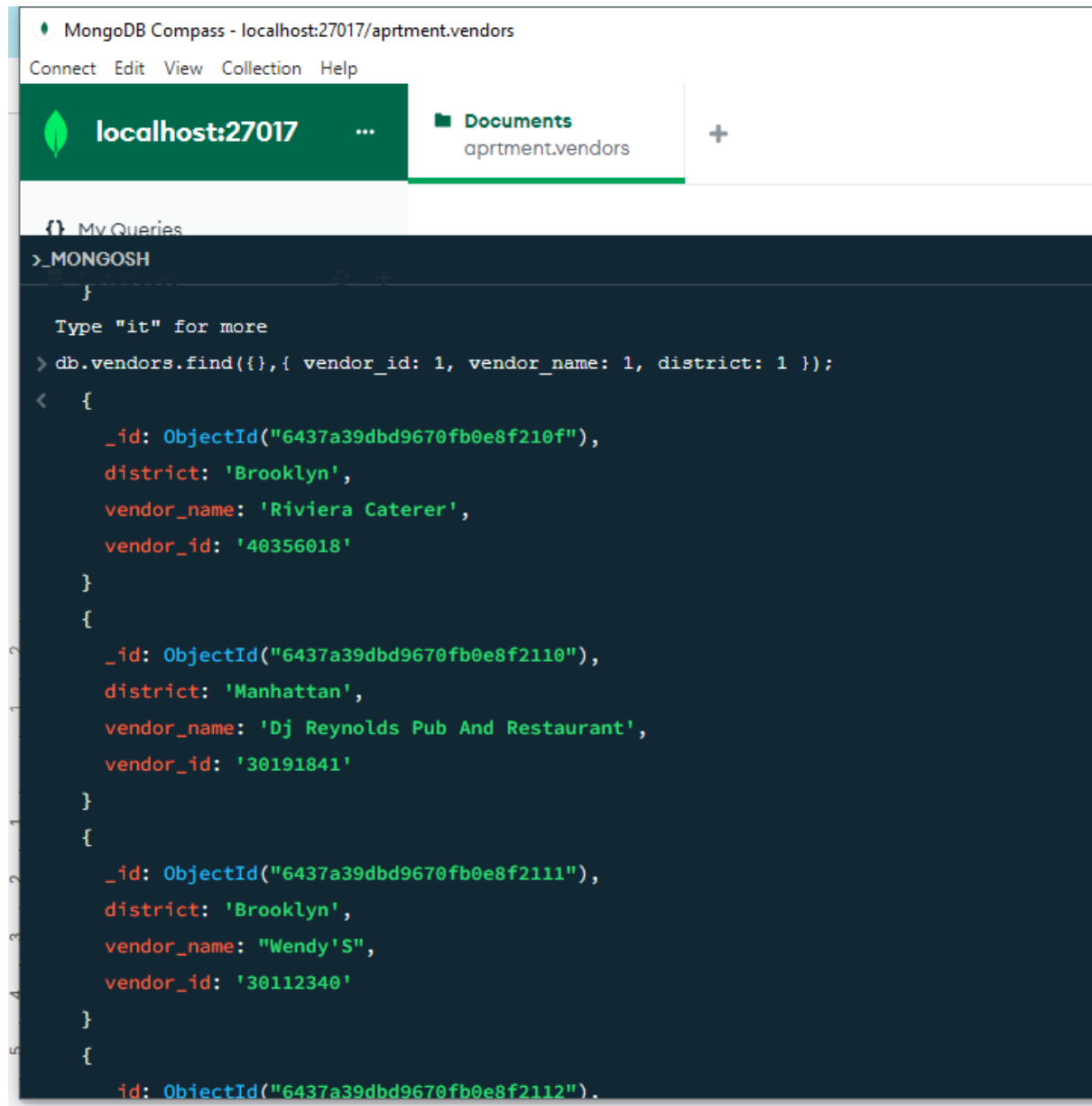


The screenshot shows the MongoDB Compass application interface. At the top, the title bar reads 'MongoDB Compass - localhost:27017/aprtment.vendors'. Below the title bar is a menu bar with 'Connect', 'Edit', 'View', 'Collection', and 'Help'. The main interface has a sidebar on the left with a green header 'localhost:27017' and a 'Documents' tab for 'aprtment.vendors'. The main area shows a document in the 'vendors' collection. The document is displayed in a dark-themed editor with a light blue background. The document structure is as follows:

```
> use aprtment
< 'switched to db aprtment'
> db.vendors.find();
< {
  _id: ObjectId("6437a39dbd9670fb0e8f210f"),
  street_address: {
    building: '2780',
    coord: [
      -73.98241999999999,
      40.579505
    ],
    street: 'Stillwell Avenue',
    zipcode: '11224'
  },
  district: 'Brooklyn',
  characteristic: 'American ',
  vendor_grades: [
    {
      date: 2014-06-10T00:00:00.000Z,
      grade: 'A',
      score: 5
    }
  ]
}
```

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 });
```

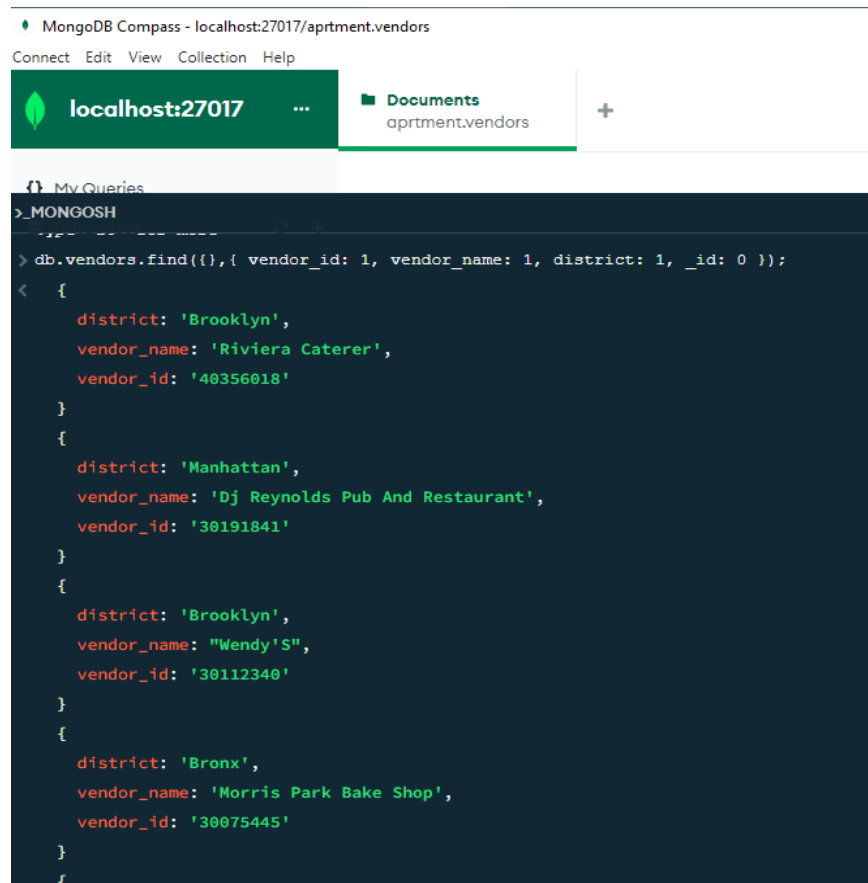


The screenshot shows the MongoDB Compass application interface. At the top, the title bar reads "MongoDB Compass - localhost:27017/aprtment.vendors". Below the title bar is a menu bar with "Connect", "Edit", "View", "Collection", and "Help". The main interface is divided into two panes. The left pane, titled "My Queries", contains a single query: `>_MONGOSH` followed by `{}`. The right pane, titled "Documents", shows the results of the query. The results are displayed as a list of four documents, each with the following fields: `_id`, `district`, `vendor_name`, and `vendor_id`. The documents are as follows:

```
{
  "_id": ObjectId("6437a39dbd9670fb0e8f210f"),
  "district": "Brooklyn",
  "vendor_name": "Riviera Caterer",
  "vendor_id": "40356018"
},
{
  "_id": ObjectId("6437a39dbd9670fb0e8f2110"),
  "district": "Manhattan",
  "vendor_name": "Dj Reynolds Pub And Restaurant",
  "vendor_id": "30191841"
},
{
  "_id": ObjectId("6437a39dbd9670fb0e8f2111"),
  "district": "Brooklyn",
  "vendor_name": "Wendy'S",
  "vendor_id": "30112340"
},
{
  "_id": ObjectId("6437a39dbd9670fb0e8f2112"),
  "district": "Manhattan",
  "vendor_name": "Dj Reynolds Pub And Restaurant",
  "vendor_id": "30191841"
}
```

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 });
```



The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'localhost:27017/aprtment.vendors'. Below the top bar, there are tabs for 'Connect', 'Edit', 'View', 'Collection', and 'Help'. The 'Documents' tab is selected, showing the collection 'aprtment.vendors'. The 'My Queries' tab is also visible. The query editor shows the following query:

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

The results are displayed as a JSON array of documents:

```
< [
  {
    district: 'Brooklyn',
    vendor_name: 'Riviera Caterer',
    vendor_id: '40356018'
  },
  {
    district: 'Manhattan',
    vendor_name: 'Dj Reynolds Pub And Restaurant',
    vendor_id: '30191841'
  },
  {
    district: 'Brooklyn',
    vendor_name: 'Wendy'S',
    vendor_id: '30112340'
  },
  {
    district: 'Bronx',
    vendor_name: 'Morris Park Bake Shop',
    vendor_id: '30075445'
  }
]
```

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

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

MongoDB Compass - localhost:27017/aprtment.vendors

Connect Edit View Collection Help



localhost:27017



Documents

aprtment.vendors



My Queries

>\_MONGOSH

}

Type "it" for more

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

< {

\_id: ObjectId("6437a39dbd9670fb0e8f210f"),

street\_address: {

building: '2780',

coord: [

-73.98241999999999,

40.579505

],

street: 'Stillwell Avenue',

zipcode: '11224'

},

district: 'Brooklyn',

characteristic: 'American ',

vendor\_grades: [

{

date: 2014-06-10T00:00:00.000Z,

grade: 'A',

score: 5

},

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

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

MongoDB Compass - localhost:27017/aprtment.vendors

Connect Edit View Collection Help

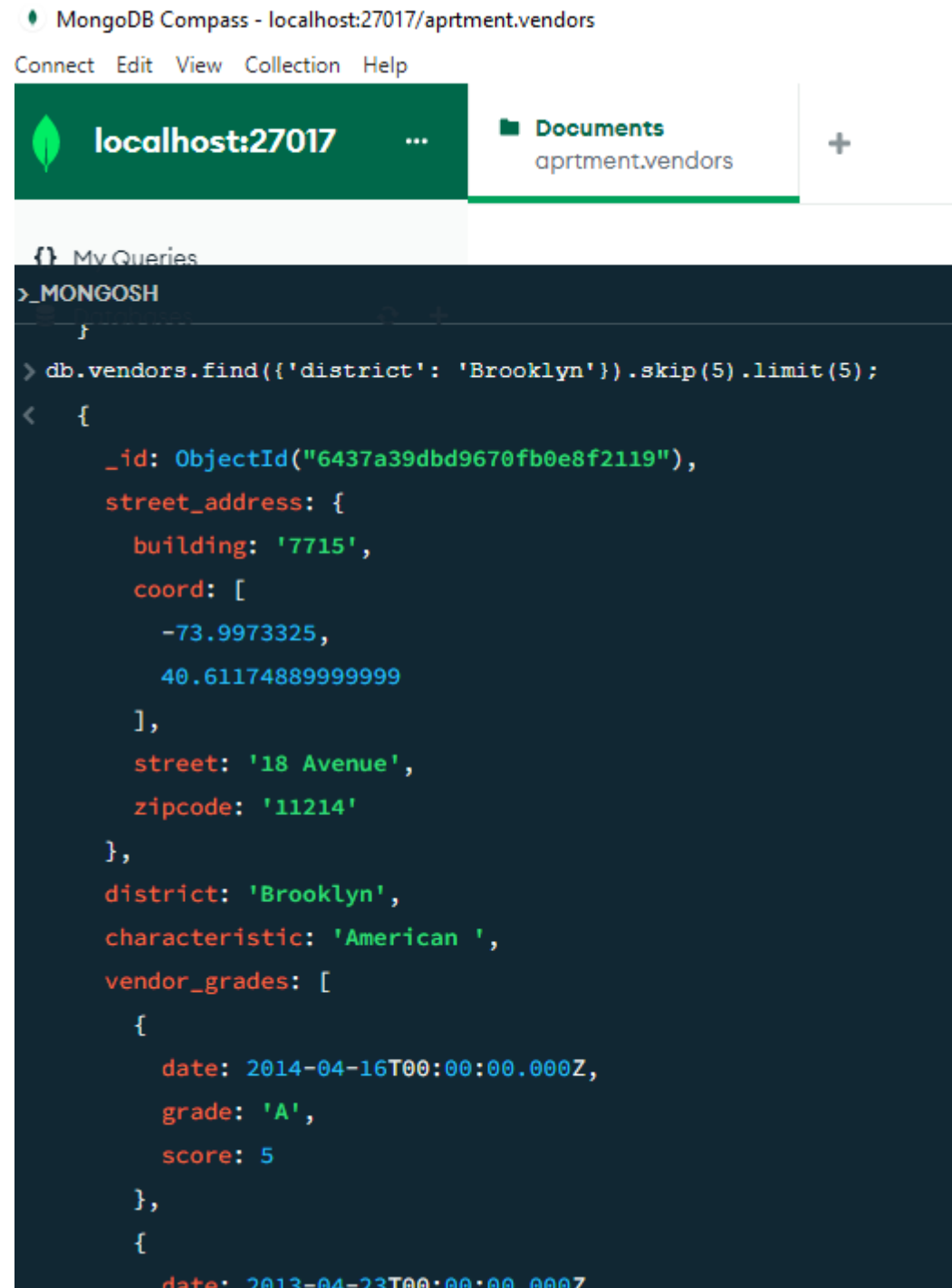
localhost:27017 ... Documents aprtment.vendors +

My Queries

```
>_MONGOSH
> db.vendors.find({'district': 'Brooklyn'}).limit(5);
< {
  _id: ObjectId("6437a39dbd9670fb0e8f210f"),
  street_address: {
    building: '2780',
    coord: [
      -73.98241999999999,
      40.579505
    ],
    street: 'Stillwell Avenue',
    zipcode: '11224'
  },
  district: 'Brooklyn',
  characteristic: 'American ',
  vendor_grades: [
    {
      date: 2014-06-10T00:00:00.000Z,
      grade: 'A',
      score: 5
    },
    {
      date: 2013-06-05T00:00:00.000Z,
```

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);
```



The screenshot shows the MongoDB Compass interface. At the top, it says 'MongoDB Compass - localhost:27017/aprtment.vendors'. Below this is a navigation bar with 'Connect', 'Edit', 'View', 'Collection', and 'Help'. The main area is divided into two panels. The left panel, titled 'localhost:27017', shows a tree view with 'Documents' and 'aprtment.vendors'. The right panel, titled 'My Queries', shows a query editor with the following query:

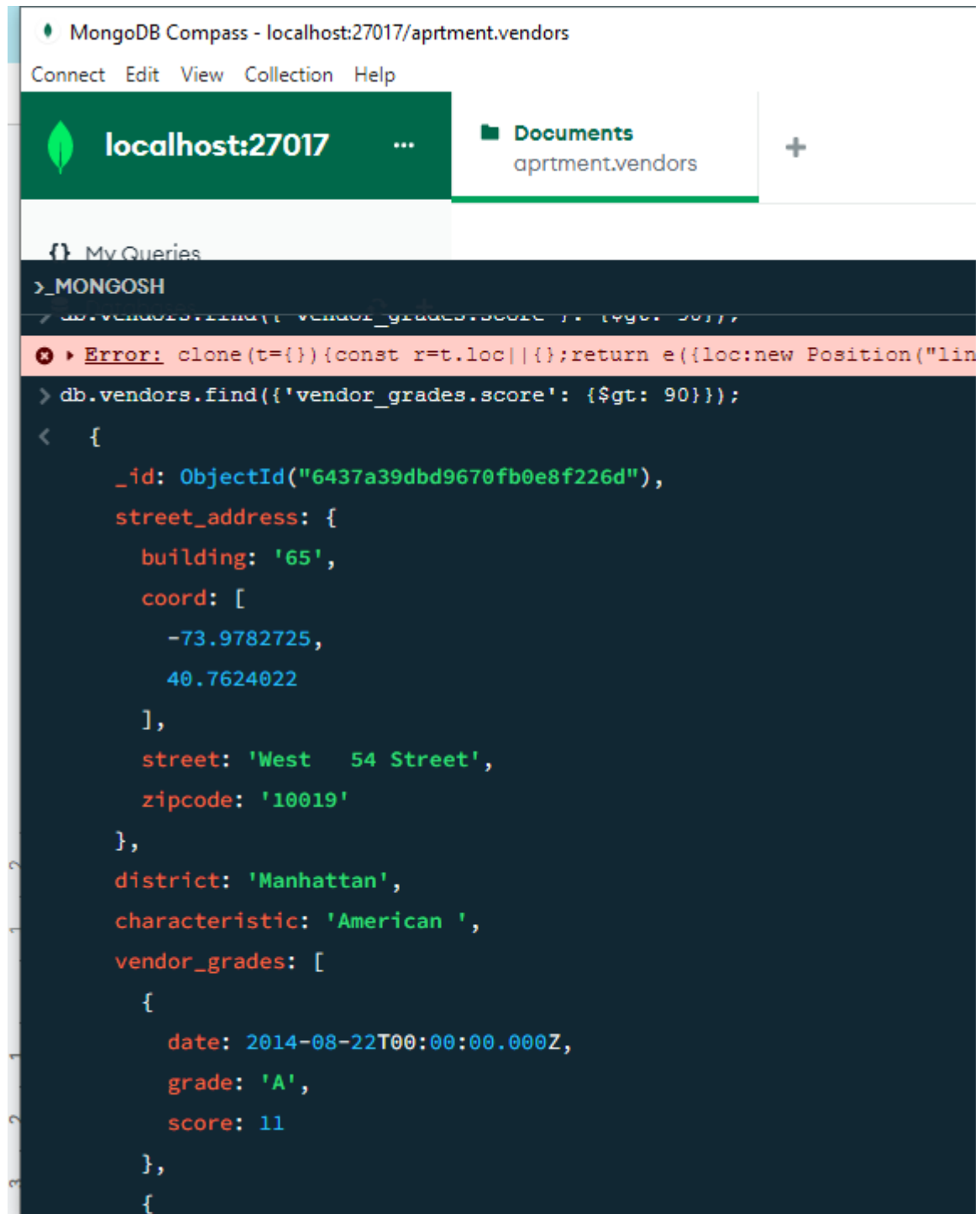
```
> db.vendors.find({'district': 'Brooklyn'}).skip(5).limit(5);
```

The result of the query is displayed in a JSON format:

```
< {
  _id: ObjectId("6437a39dbd9670fb0e8f2119"),
  street_address: {
    building: '7715',
    coord: [
      -73.9973325,
      40.611748899999999
    ],
    street: '18 Avenue',
    zipcode: '11214'
  },
  district: 'Brooklyn',
  characteristic: 'American ',
  vendor_grades: [
    {
      date: 2014-04-16T00:00:00.000Z,
      grade: 'A',
      score: 5
    },
    {
      date: 2013-04-23T00:00:00.000Z,
```

8. find the vendors who achieved a score **more than 90** in **vendor\_grades**

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



The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'localhost:27017/aprtment.vendors'. The left sidebar shows the 'Documents' tab for the 'aprtment.vendors' collection. The main area displays a query in the 'My Queries' tab:

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

An error message is shown below the query:

```
✖ Error: clone(t={}) {const r=t.loc||{};return e({loc:new Position("lin
```

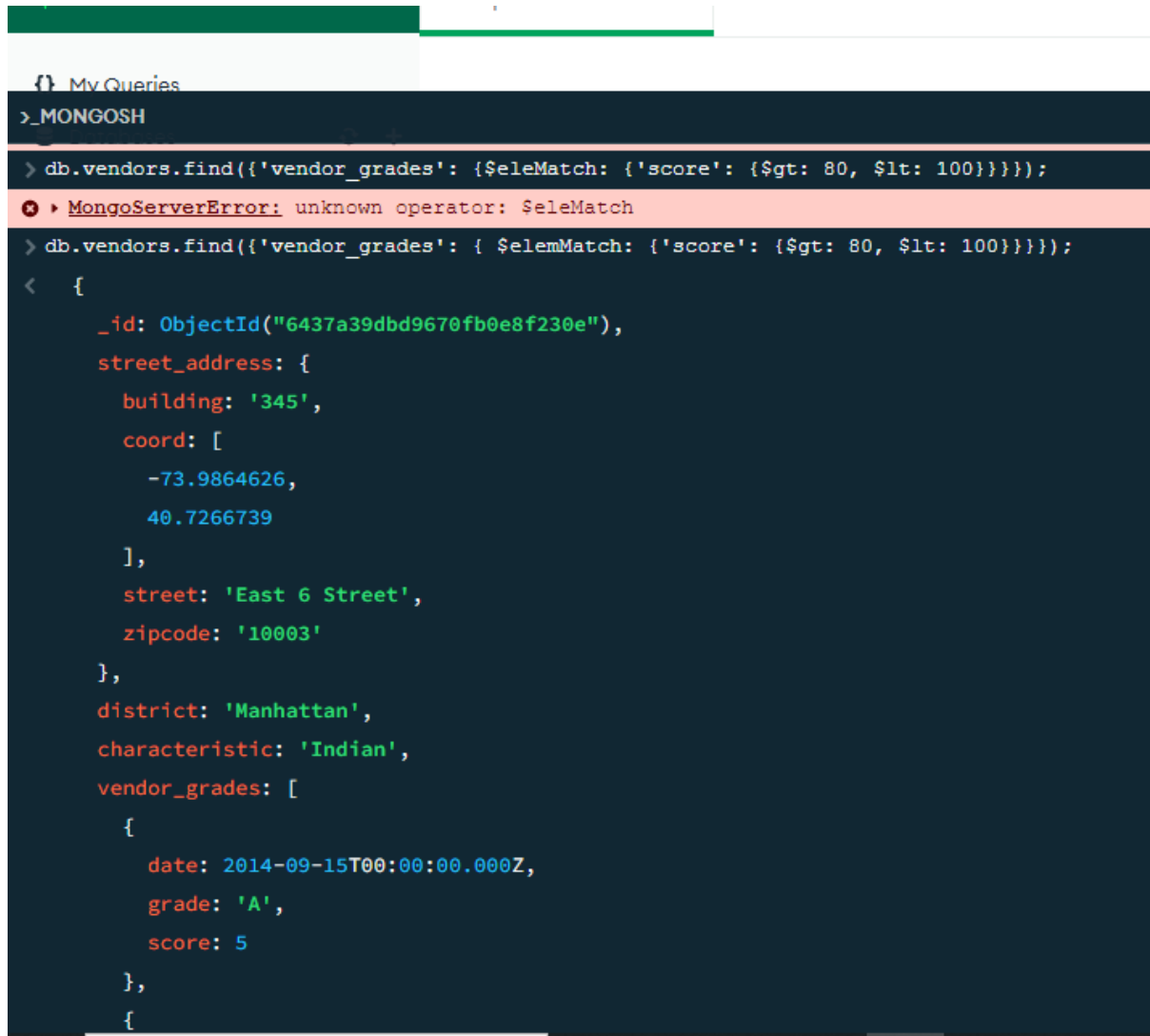
The query result is displayed below the error message:

```
> db.vendors.find({'vendor_grades.score': {$gt: 90}});
< {
  _id: ObjectId("6437a39dbd9670fb0e8f226d"),
  street_address: {
    building: '65',
    coord: [
      -73.9782725,
      40.7624022
    ],
    street: 'West 54 Street',
    zipcode: '10019'
  },
  district: 'Manhattan',
  characteristic: 'American ',
  vendor_grades: [
    {
      date: 2014-08-22T00:00:00.000Z,
      grade: 'A',
      score: 11
    },
    {
```



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 }}}});
```

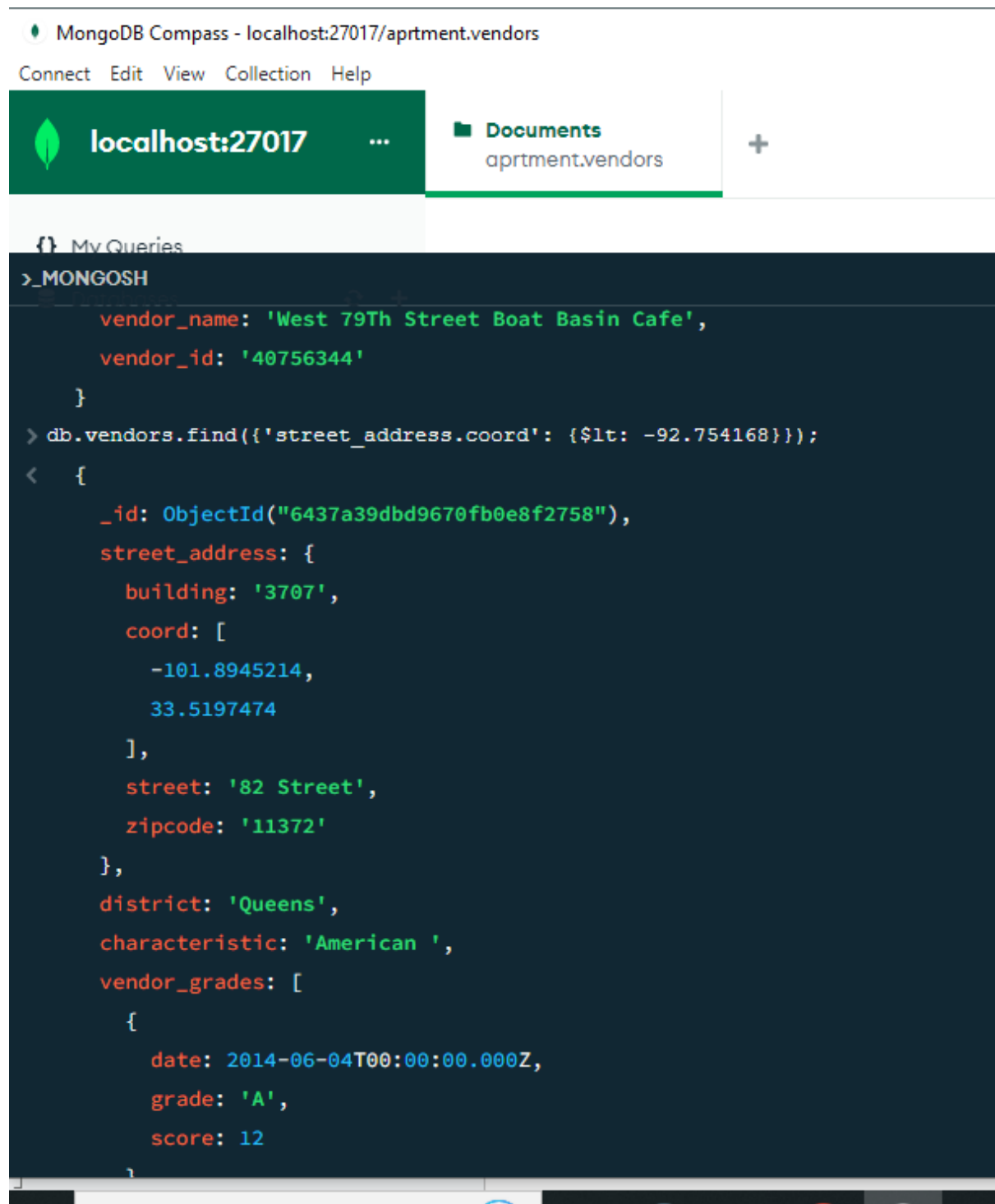


The screenshot shows a MongoDB shell interface with a dark background. At the top, there is a tab labeled 'My Queries'. Below it, the prompt '>\_MONGOSH' is visible. The user has entered the query: `> db.vendors.find({'vendor_grades': { $elemMatch: {'score': { $gt: 80, $lt: 100 }}}});`. An error message is displayed: `✖ ▶ MongoServerError: unknown operator: $elemMatch`. The user then enters the corrected query: `> db.vendors.find({'vendor_grades': { $elemMatch: {'score': { $gt: 80, $lt: 100 }}}});`. The result is shown as a JSON document: 

```
< {
  _id: ObjectId("6437a39dbd9670fb0e8f230e"),
  street_address: {
    building: '345',
    coord: [
      -73.9864626,
      40.7266739
    ],
    street: 'East 6 Street',
    zipcode: '10003'
  },
  district: 'Manhattan',
  characteristic: 'Indian',
  vendor_grades: [
    {
      date: 2014-09-15T00:00:00.000Z,
      grade: 'A',
      score: 5
    },
    {
```

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

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

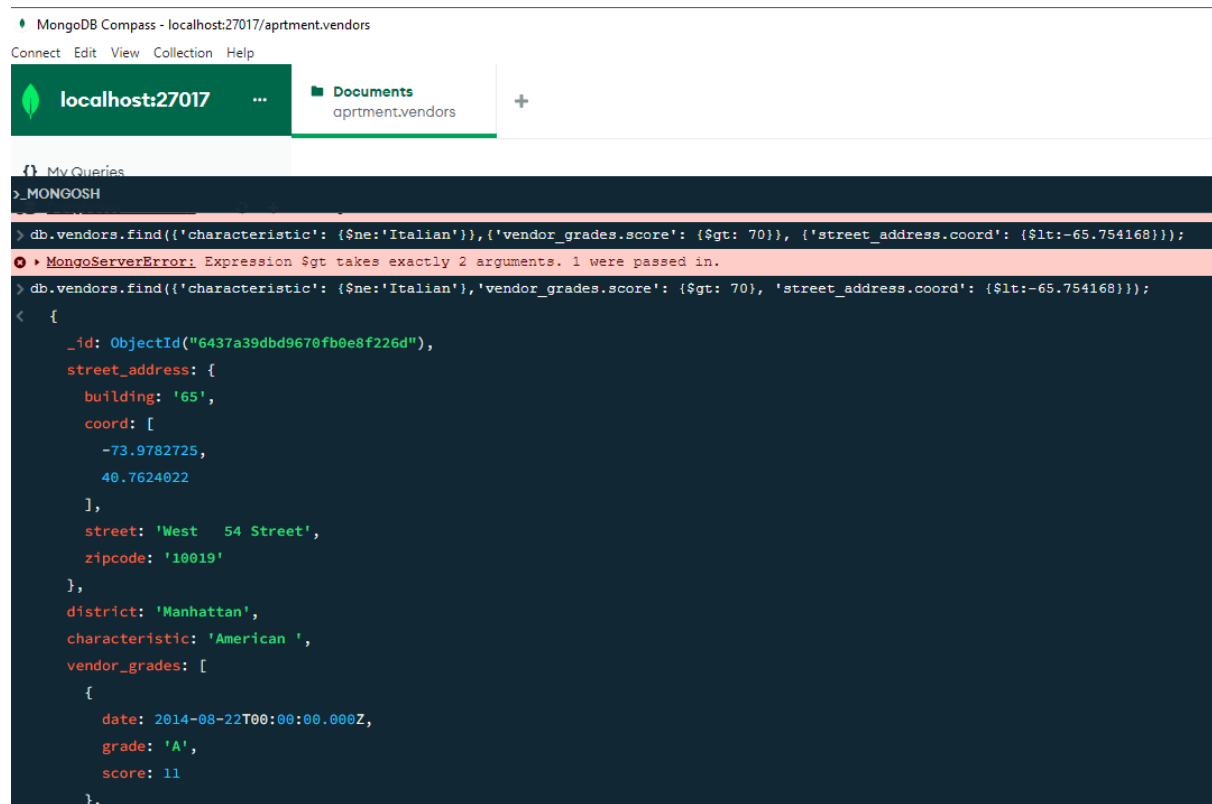


The screenshot shows the MongoDB Compass interface. At the top, the title bar reads 'MongoDB Compass - localhost:27017/aprtment.vendors'. Below it are tabs for 'Connect', 'Edit', 'View', 'Collection', and 'Help'. The main interface has a green header with a leaf icon, 'localhost:27017', and a 'Documents' tab for 'aprtment.vendors'. Below the header, there's a 'My Queries' section. The main area is a dark terminal window with a MONGOSH prompt. It shows a query: `db.vendors.find({'street_address.coord': {'$lt: -92.754168}});` and its result as a JSON document.

```
>_MONGOSH
  vendor_name: 'West 79Th Street Boat Basin Cafe',
  vendor_id: '40756344'
}
> db.vendors.find({'street_address.coord': {'$lt: -92.754168}});
< {
  _id: ObjectId("6437a39dbd9670fb0e8f2758"),
  street_address: {
    building: '3707',
    coord: [
      -101.8945214,
      33.5197474
    ],
    street: '82 Street',
    zipcode: '11372'
  },
  district: 'Queens',
  characteristic: 'American ',
  vendor_grades: [
    {
      date: 2014-06-04T00:00:00.000Z,
      grade: 'A',
      score: 12
    }
  ]
}
```

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}});
```

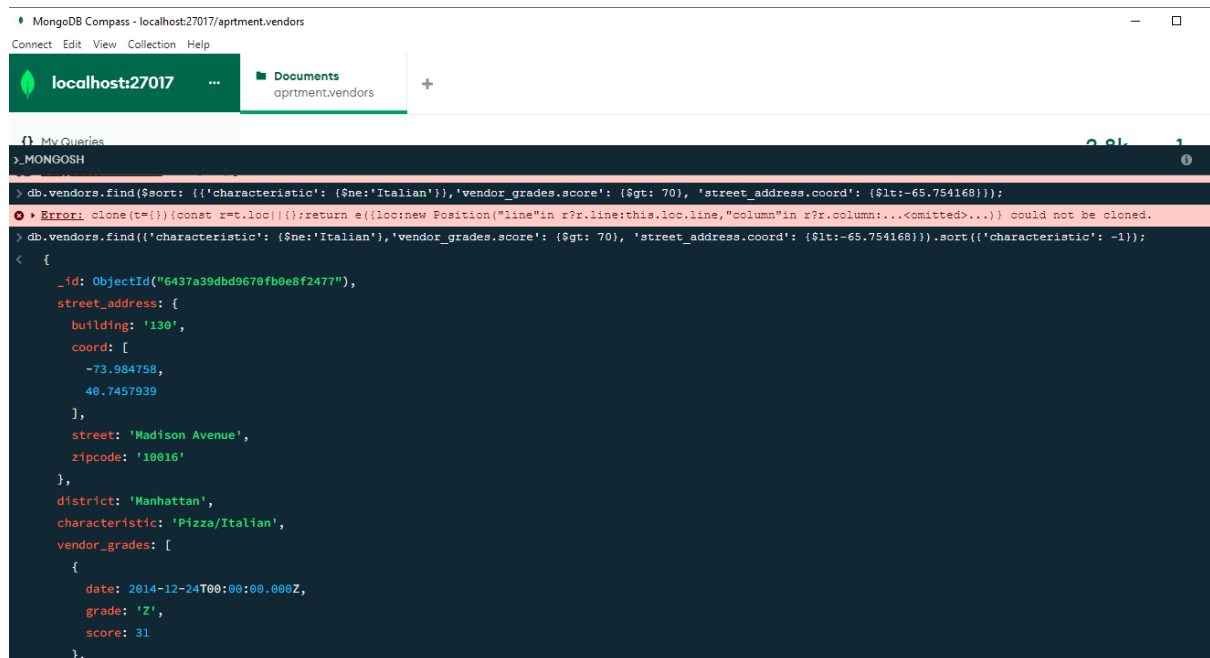


The screenshot shows the MongoDB Compass interface. At the top, it says 'MongoDB Compass - localhost:27017/aprtment.vendors'. Below that is a menu bar with 'Connect', 'Edit', 'View', 'Collection', and 'Help'. The main area has a green sidebar with 'localhost:27017' and a 'Documents' tab for 'aprtment.vendors'. The central pane shows a query in the 'My Queries' section. The query is: `db.vendors.find({'characteristic': {$ne:'Italian'}}, {'vendor_grades.score': {$gt: 70}}, {'street_address.coord': {$lt:-65.754168}});`. Below the query, there is a red error message: 'MongoServerError: Expression \$gt takes exactly 2 arguments. 1 were passed in.' Below the error, the same query is shown again, but with a different syntax: `db.vendors.find({'characteristic': {$ne:'Italian'}, 'vendor_grades.score': {$gt: 70}, 'street_address.coord': {$lt:-65.754168}});`. The result of the query is shown as a single document in a dark blue background. The document has the following fields: `_id` (ObjectId), `street_address` (a nested object with `building` and `coord`), `street`, `zipcode`, `district`, `characteristic`, and `vendor_grades` (an array of objects with `date`, `grade`, and `score`).

```
> db.vendors.find({'characteristic': {$ne:'Italian'}}, {'vendor_grades.score': {$gt: 70}}, {'street_address.coord': {$lt:-65.754168}});
MongoServerError: Expression $gt takes exactly 2 arguments. 1 were passed in.
> db.vendors.find({'characteristic': {$ne:'Italian'}, 'vendor_grades.score': {$gt: 70}, 'street_address.coord': {$lt:-65.754168}});
{
  _id: ObjectId("6437a39dbd9670fb0e8f226d"),
  street_address: {
    building: '65',
    coord: [
      -73.9782725,
      40.7624022
    ],
  },
  street: 'West 54 Street',
  zipcode: '10019',
},
district: 'Manhattan',
characteristic: 'American ',
vendor_grades: [
  {
    date: 2014-08-22T00:00:00.000Z,
    grade: 'A',
    score: 11
  },
]
```

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});
```



The screenshot shows the MongoDB Compass application. The top bar indicates the connection to 'localhost:27017/aprtment.vendors'. Below the top bar, there's a 'Documents' tab for 'aprtment.vendors'. The main area displays a query in the 'My Queries' tab:

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

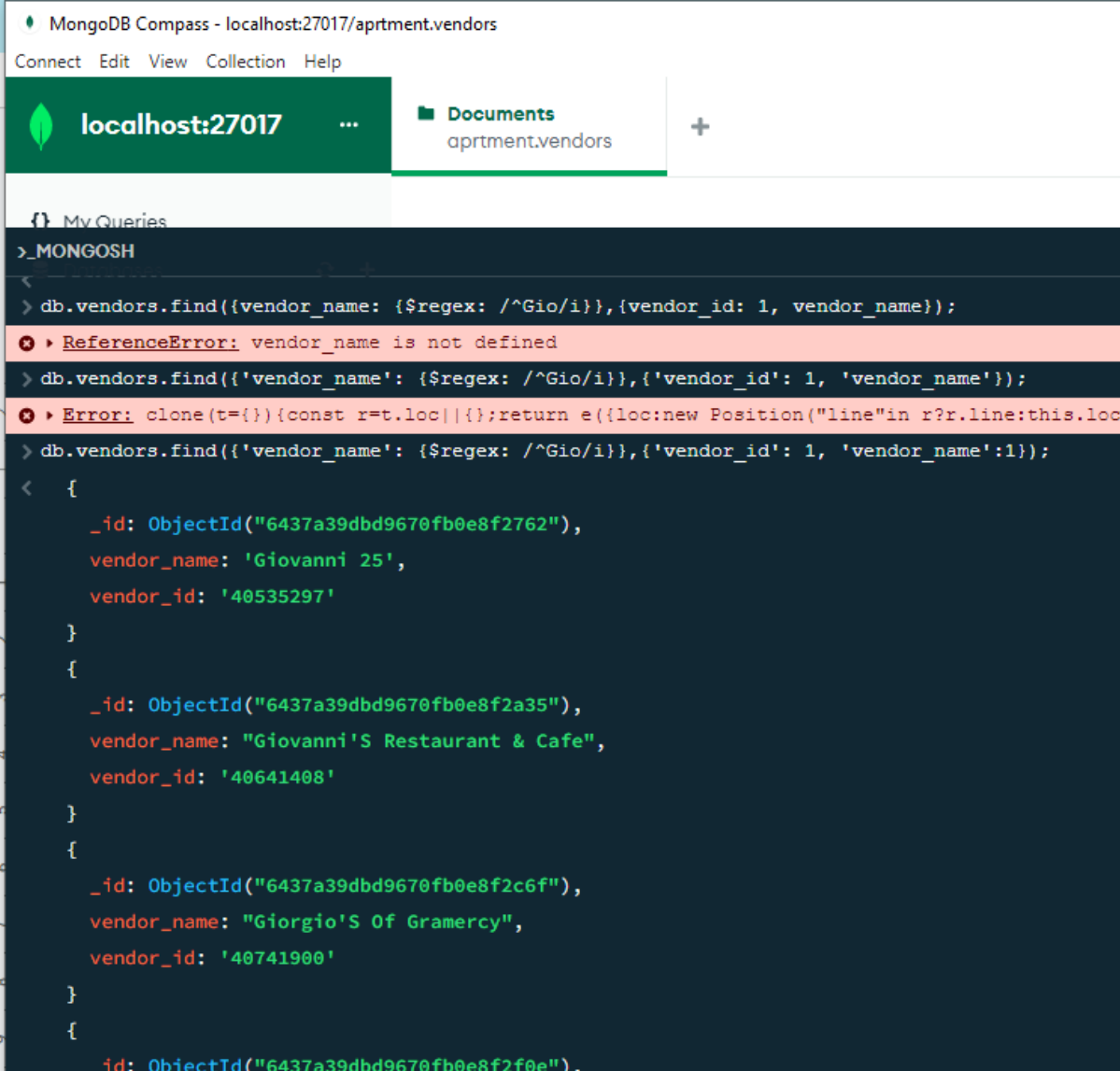
An error message is visible: "Error: clone(t={}) (const r=t.loc||{}):return e(({loc:new Position("line" in r?r.line:this.loc.line,"column" in r?r.column:...<omitted>...))} could not be cloned."

The query result is displayed as a JSON document:

```
< {
  _id: ObjectId("6437a39dbd9670fb0e8f2477"),
  street_address: {
    building: '130',
    coord: [
      -73.984758,
      40.7457939
    ],
    street: 'Madison Avenue',
    zipcode: '10016'
  },
  district: 'Manhattan',
  characteristic: 'Pizza/Italian',
  vendor_grades: [
    {
      date: 2014-12-24T00:00:00.000Z,
      grade: 'Z',
      score: 31
    }
  ],
}
```

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});
```



The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'localhost:27017/aprtment.vendors'. The left sidebar shows the 'Documents' tab for the 'aprtment.vendors' collection. The main area displays a query in the 'My Queries' tab:

```
>_MONGOSH
> db.vendors.find({'vendor_name': {$regex: /^Gio/i}},{'vendor_id': 1, vendor_name});
✖ ReferenceError: vendor_name is not defined
> db.vendors.find({'vendor_name': {$regex: /^Gio/i}},{'vendor_id': 1, 'vendor_name'});
✖ Error: clone(t={}){const r=t.loc||{};return e({loc:new Position("line"in r?r.line:this.loc
> db.vendors.find({'vendor_name': {$regex: /^Gio/i}},{'vendor_id': 1, 'vendor_name':1});
<
{
  _id: ObjectId("6437a39dbd9670fb0e8f2762"),
  vendor_name: 'Giovanni 25',
  vendor_id: '40535297'
}
{
  _id: ObjectId("6437a39dbd9670fb0e8f2a35"),
  vendor_name: 'Giovanni'S Restaurant & Cafe',
  vendor_id: '40641408'
}
{
  _id: ObjectId("6437a39dbd9670fb0e8f2c6f"),
  vendor_name: 'Giorgio'S Of Gramercy',
  vendor_id: '40741900'
}
{
  _id: ObjectId("6437a39dbd9670fb0e8f2f0e"),
```

The query results show four documents from the 'vendors' collection, each with an '\_id', 'vendor\_name', and 'vendor\_id' field. The vendor names are 'Giovanni 25', 'Giovanni'S Restaurant & Cafe', 'Giorgio'S Of Gramercy', and an incomplete entry.

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});
```

```
<
> db.vendors.find({'vendor_name': {$regex: 'cas$'}},{'vendor_id': 1, 'vendor_name':1});
< {
  _id: ObjectId("6437a39dbd9670fb0e8f2c0d"),
  vendor_name: 'Cositas Ricas',
  vendor_id: '40729439'
}
aprtment> |
```

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});
```

```
    }  
> db.vendors.find({'vendor_name': {'$regex': 'Rpg'}}, {'vendor_id': 1, 'vendor_name': 1});  
<  
> db.vendors.find({'vendor_name': {'$regex': '/Rpg/i'}}, {'vendor_id': 1, 'vendor_name': 1});  
<  
aprtment> |
```

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

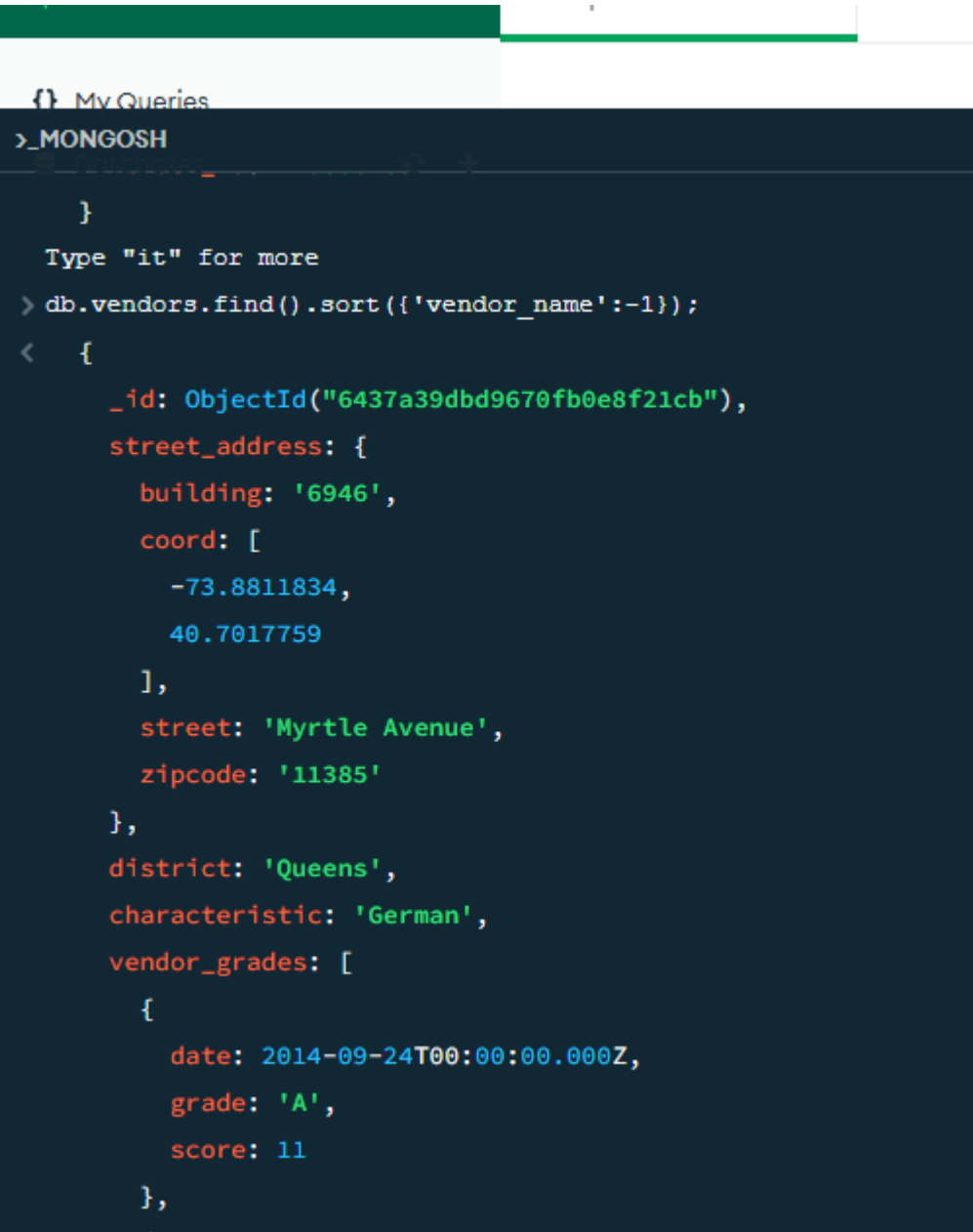
```
db.vendors.find().sort({'vendor_name': 1});
```

```
>_MONGOSH
<
> db.vendors.find().sort({'vendor_name':1});
< {
  _id: ObjectId("6437a39dbd9670fb0e8f2d9f"),
  street_address: {
    building: '129',
    coord: [
      -73.962943,
      40.685007
    ],
    street: 'Gates Avenue',
    zipcode: '11238'
  },
  district: 'Brooklyn',
  characteristic: 'Italian',
  vendor_grades: [
    {
      date: 2014-03-06T00:00:00.000Z,
      grade: 'A',
      score: 5
    },
    {
      date: 2013-08-29T00:00:00.000Z,
```

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



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



The screenshot shows a MongoDB shell window with a dark background. At the top, there's a tab labeled 'My Queries'. Below it, the prompt '> \_MONGOSH' is visible. The user has entered the command `db.vendors.find().sort({'vendor_name':-1});`. The shell has responded with a JSON document representing a vendor. The document includes fields for `_id`, `street_address` (which is an object containing `building`, `coord`, and `street`), `zipcode`, `district`, `characteristic`, and `vendor_grades` (an array of objects). The `coord` array contains two numbers: -73.8811834 and 40.7017759. The `date` field in the vendor\_grades array is in ISO 8601 format.

```
}
Type "it" for more
> db.vendors.find().sort({'vendor_name':-1});
< {
  _id: ObjectId("6437a39dbd9670fb0e8f21cb"),
  street_address: {
    building: '6946',
    coord: [
      -73.8811834,
      40.7017759
    ],
    street: 'Myrtle Avenue',
    zipcode: '11385'
  },
  district: 'Queens',
  characteristic: 'German',
  vendor_grades: [
    {
      date: 2014-09-24T00:00:00.000Z,
      grade: 'A',
      score: 11
    },
  ],
}
```

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

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

```
{ } My Queries
>_MONGOSH
Type "it" for more
> db.vendors.find({'street_address.street': {$exists : true}});
< {
  _id: ObjectId("6437a39dbd9670fb0e8f210f"),
  street_address: {
    building: '2780',
    coord: [
      -73.982419999999999,
      40.579505
    ],
    street: 'Stillwell Avenue',
    zipcode: '11224'
  },
  district: 'Brooklyn',
  characteristic: 'American ',
  vendor_grades: [
    {
      date: 2014-06-10T00:00:00.000Z,
      grade: 'A',
      score: 5
    },
    {
      date: 2013-06-05T00:00:00.000Z,
```

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'}});
```

```
> db.vendors.find({'street_address.coord': {'$type': 'double'}});  
< {  
  _id: ObjectId("6437a39dbd9670fb0e8f210f"),  
  street_address: {  
    building: '2780',  
    coord: [  
      -73.98241999999999,  
      40.579505  
    ],  
    street: 'Stillwell Avenue',  
    zipcode: '11224'  
  },  
  district: 'Brooklyn',  
  characteristic: 'American ',  
  vendor_grades: [  
    {  
      date: 2014-06-10T00:00:00.000Z,  
      grade: 'A',  
      score: 5  
    },  
    {
```

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" } ]}});
```

```
My Queries
> MONGODB

Error: clone(t={}) (const r=t.loc||{});return e({loc:new Position("line" in r?r.line:this.loc.line,"column" in r?r.column:...<omitted>...)} could not be cloned.
> db.vendors.find({$and: [{ 'vendor_grades.score': { $lt: 5 } }, { 'characteristic': { $ne: 'Italian' } }, { $or: [{ 'district': 'Manhattan' }, { 'district': 'Brooklyn' }] } ]})
Error: clone(t={}) (const r=t.loc||{});return e({loc:new Position("line" in r?r.line:this.loc.line,"column" in r?r.column:...<omitted>...)} could not be cloned.
> db.vendors.find({$and: [{ 'vendor_grades.score': { $lt: 5 } }, { 'characteristic': { $ne: 'Italian' } }, { $or: [{ 'district': 'Manhattan' }, { 'district': 'Brooklyn' }] } ]})
Error: clone(t={}) (const r=t.loc||{});return e({loc:new Position("line" in r?r.line:this.loc.line,"column" in r?r.column:...<omitted>...)} could not be cloned.
> db.vendors.find({$and: [{ "vendor_grades.score": { $lt: 5 } }, { "characteristic": { $ne: "Italian" } }, { $or: [{ "district": "Manhattan" }, { "district": "Brooklyn" } ] } ]})
{
  _id: ObjectId("6437a39dbd9670fb0e8f2110"),
  street_address: {
    building: '351',
    coord: [
      -73.98513559999999,
      40.7676919
    ],
    street: 'West 57 Street',
    zipcode: '10019'
  },
  district: 'Manhattan',
  characteristic: 'Irish',
  vendor_grades: [
    {
      date: 2014-09-06T00:00:00.000Z,
```