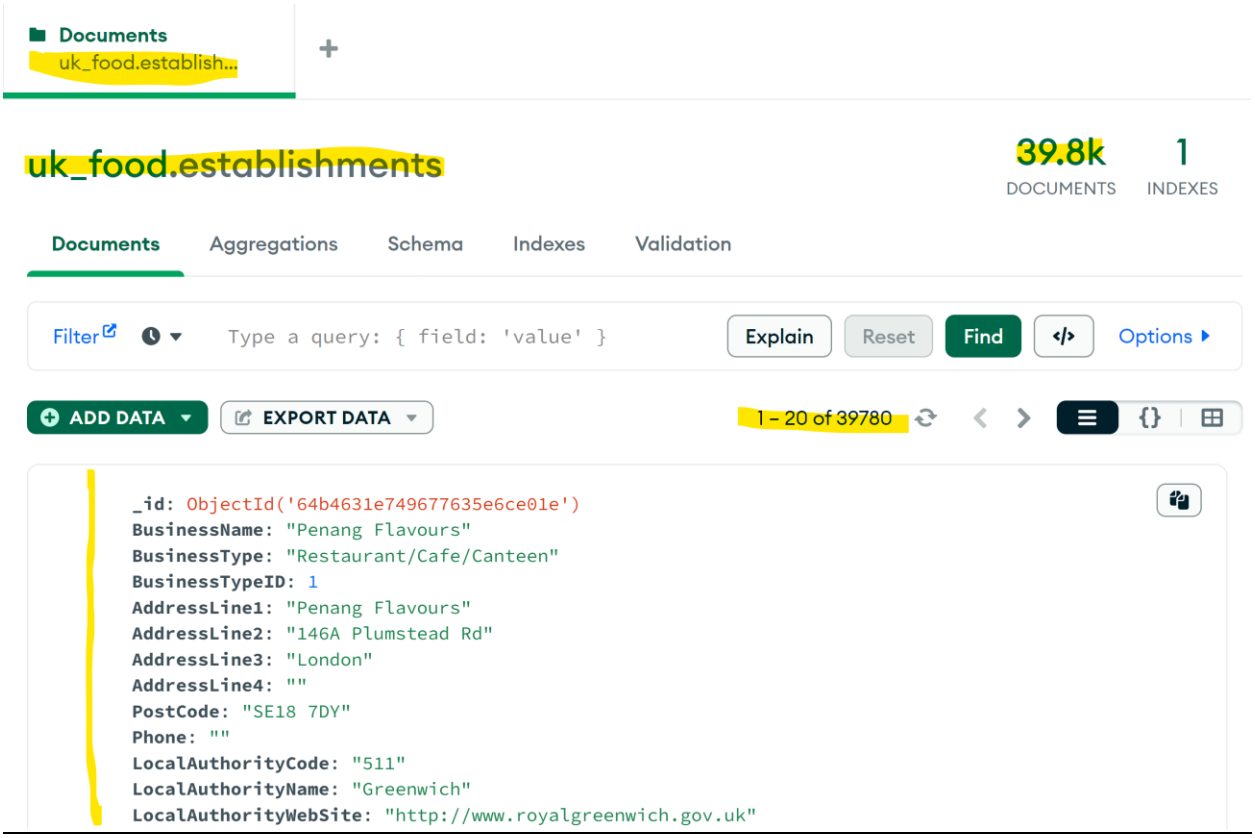
Module 12 nosql-challenge - screen shots v1
Raj Agrawal / SMU DS / July 2023
===

Deliverable 1: Part 1: Database and Jupyter Notebook Set Up



Part 1: Database and Jupyter Notebook Set Up

##This step is established----------

Import the data provided in the `establishments.json` file from your Terminal. Name the database `uk_food` and the collection `establishments`.

Within this markdown cell, copy the line of text you used to import the data from your Terminal. This way, future analysts will be able to repeat your process.

e.g.: Import the dataset with `mongoimport --type json -d uk_food -c establishments --drop --jsonArray establishments.json`

##attached screenshot docs - <<Module 12 nosql-challenge - screen shots v1>>

In [47]:
```
# Import dependencies
from pymongo import MongoClient
from pprint import pprint
import pandas as pd
```

In [48]:
```
# Create an instance of MongoClient
mongo = MongoClient(port=27017)
```

In [49]:
```
# confirm that our new database was created
print(mongo.list_database_names())
```

```
['admin', 'config', 'local', 'met', 'uk_food']
```

In [50]:
```
# assign the uk_food database to a variable name
db = mongo['uk_food']
```

In [51]:
```
# review the collections in our new database
print(db.list_collection_names())
```

```
['establishments']
```

LocalAuthorityName as "Dover"

In [41]:
```
# Find how many documents have LocalAuthorityName as "Dover"
query = {"LocalAuthorityName": "Dover"}

# Cast the results as a list and save the results to a variable
data = establishments.find(query)

# Pretty print the results
df = pd.DataFrame(data)
print(df.shape)
df.head()
```

```
(994, 28)
```

Out[41]:

	_id	FHRSID	ChangesByServerID	LocalAuthorityBusinessID	BusinessName	BusinessType	BusinessTypeID	AddressLine
0	64b469d3fb4578cd17f8c216	254719	0	PI/000069980	Refreshment Kiosk	Restaurant/Cafe/Canteen	1	The Ba
1	64b469d3fb4578cd17f8c217	1034540	0	PI/000078691	The Coastguard Inn	Pub/bar/nightclub	7843	The Ba
2	64b469d3fb4578cd17f8c218	254250	0	PI/000066174	The Pines Calyx	Other catering premises	7841	The Pine Garde
3	64b469d3fb4578cd17f8c219	551803	0	PI/000070948	The Tea Room	Restaurant/Cafe/Canteen	1	The Pine Garde
4	64b469d3fb4578cd17f8c21a	632212	0	PI/000043474	Lenox House	Hotel/bed & breakfast/guest house	7842	27 Granvill Roa

5 rows × 28 columns

Result results of "LocalAuthorityName": "Dover"}
(994, 28)
is saved as a screen shot doc----------<<Module 12 nosql-challenge - screen shots v1>>
Before running next step command for -----Deleteting

In [68]:
```
# Delete all documents where LocalAuthorityName is "Dover"
query = {"LocalAuthorityName": "Dover"}
establishments.delete_many(query)
```

Out[68]:
```
<pymongo.results.DeleteResult at 0x2164cd42c80>
```

In [69]:
```
# Check if any remaining documents include Dover
results = establishments.find(query)

for result in results:
    print(result)
```

In [62]:
```
# Check that other documents remain with 'find_one'
establishments.find_one()
```

Out[62]:
```
{'_id': ObjectId('64b4631e749677635e6ce01e'),
 'BusinessName': 'Penang Flavours',
 'BusinessType': 'Restaurant/Cafe/Canteen',
 'BusinessTypeID': 1,
 'AddressLine1': 'Penang Flavours',
 'AddressLine2': '146A Plumstead Rd',
 'AddressLine3': 'London',
 'AddressLine4': '',
 'PostCode': 'SE18 7DY',
 'Phone': '',
 'LocalAuthorityCode': '511',
 'LocalAuthorityName': 'Greenwich',
 'LocalAuthorityWebSite': 'http://www.royalgreenwich.gov.uk',
 'LocalAuthorityEmailAddress': 'health@royalgreenwich.gov.uk',
 'scores': {'Hygiene': '', 'Structural': '', 'ConfidenceInManagement': ''},
```

Deliverable 2: Part 2: Update the Database

In [54]:
```
# Create a dictionary for the new restaurant data
new_restaurant = {
    "BusinessName":"Penang Flavours",
    "BusinessType":"Restaurant/Cafe/Canteen",
    "BusinessTypeID":"",
    "AddressLine1":"Penang Flavours",
    "AddressLine2":"146A Plumstead Rd",
    "AddressLine3":"London",
    "AddressLine4":"",
    "PostCode":"SE18 7DY",
    "Phone":"",
    "LocalAuthorityCode":"511",
    "LocalAuthorityName":"Greenwich",
    "LocalAuthorityWebSite":"http://www.royalgreenwich.gov.uk",
    "LocalAuthorityEmailAddress":"health@royalgreenwich.gov.uk",
    "scores":{
        "Hygiene":"",
        "Structural":"",
        "ConfidenceInManagement":""
    },
    "SchemeType":"FHRS",
    "geocode":{
        "longitude":"0.08384000",
        "latitude":"51.49014200"
    },
    "RightToReply":"",
    "Distance":4623.9723280747176,
    "NewRatingPending":True
}
```

In [55]:
```
# Insert the new restaurant into the collection
db.establishments.insert_one(new_restaurant)
```

Out[55]:
```
<pymongo.results.InsertOneResult at 0x2164e3a7550>
```

In [57]:
```
# Find the BusinessTypeID for "Restaurant/Cafe/Canteen" and return only the BusinessTypeID and BusinessType fields
query = {"BusinessType": "Restaurant/Cafe/Canteen"}
fields = {"BusinessTypeID":1, "BusinessType":1}
limit = 1

# Cast the results as a list and save the results to a variable
data = establishments.find(query, fields).limit(limit)

# Pretty print the results
df = pd.DataFrame(data)
df.head()
```

Out[57]:

	_id	BusinessType	BusinessTypeID
0	64b4631e749677635e6ce01e	Restaurant/Cafe/Canteen	1

3. Update the new restaurant with the `BusinessTypeID` you found.

In [58]:
```
# Update the new restaurant with the correct BusinessTypeID
establishments.update_one({'BusinessName': 'Penang Flavours'}, {'$set': {'BusinessTypeID': 1}})
```

Out[58]:
```
<pymongo.results.UpdateResult at 0x216496392d0>
```

In [59]:
```
# Confirm that the new restaurant was updated
query = {'BusinessName': 'Penang Flavours'}
results = db.establishments.find(query)
for result in results:
    pprint(result)
```

```
{'AddressLine1': 'Penang Flavours',
 'AddressLine2': '146A Plumstead Rd',
 'AddressLine3': 'London',
 'AddressLine4': '',
 'BusinessName': 'Penang Flavours',
 'BusinessType': 'Restaurant/Cafe/Canteen',
 'BusinessTypeID': 1,
 'Distance': 4633.973328074718
```

Deliverable 3: Part 3: Exploratory Analysis

1. Which establishments have a hygiene score equal to 20?

In [35]:
```
# Find the establishments with a hygiene score of 20
query = {"scores.Hygiene": 20}
fields = {"FHRSID":1, "BusinessName":1, "BusinessType":1, "scores.Hygiene":1}
sort = [('BusinessName', 1)]

# Cast the results as a list and save the results to a variable
results = establishments.find(query, fields).sort(sort)

# Use count_documents to display the number of documents in the result
nums = establishments.count_documents(query)
print(f"There are {nums} businesses with a hygiene rating = 20\n")

# Display the first document in the results using pprint
for result in results:
    pprint(result)
    break
```

```
There are 41 businesses with a hygiene rating = 20

{'BusinessName': 'A1 News & Wine',
 'BusinessType': 'Retailers - other',
 'FHRSID': 570096,
 '_id': ObjectId('64b469dafb4578cd17f94658'),
 'scores': {'Hygiene': 20}}
```

2. Which establishments in London have a `RatingValue` greater than or equal to 4?

In [37]:
```
# Find the establishments with London as the Local Authority and has a RatingValue greater than or equal to 4.
query = {"LocalAuthorityName":{"$regex": "London"},
         "RatingValue": {"$gte": 4}}

fields = {"FHRSID":1, "BusinessName":1, "BusinessType":1, "LocalAuthorityName":1, "RatingValue":1}
sort = [('RatingValue', -1)]

# Cast the results as a list and save the results to a variable
results = establishments.find(query, fields).sort(sort)

# Use count_documents to display the number of documents in the result
nums = establishments.count_documents(query)
print(f"There are {nums} businesses matching the query\n")

# Display the first document in the results using pprint
for result in results:
    pprint(result)
    break
```

```
There are 33 businesses matching the query

{'BusinessName': 'Mv City Cruises Erasmus',
 'BusinessType': 'Other catering premises',
 'FHRSID': 1130836,
 'LocalAuthorityName': 'City of London Corporation',
 'RatingValue': 5,
 '_id': ObjectId('64b469d6fb4578cd17f8fd70')}
```

In [41]:
```
# Convert result to Pandas DataFrame
results = establishments.find(query, fields).sort(sort).limit(limit)
df = pd.DataFrame(results)
print(df.shape)
df.head(10)
```

```
(5, 7)
```

Out[41]:

	_id	FHRSID	BusinessName	BusinessType	RatingValue	scores	geocode
0	64b469d9fb4578cd17f9388e	694609	Volunteer	Pub/bar/nightclub	5	{'Hygiene': 0}	{'longitude': 0.09208, 'latitude': 51.4873437}
1	64b469d9fb4578cd17f938a6	695241	Plumstead Manor Nursery	Caring Premises	5	{'Hygiene': 0}	{'longitude': 0.0859939977526665, 'latitude': ...
2	64b469d9fb4578cd17f93861	695223	Iceland	Retailers - supermarkets/hypermarkets	5	{'Hygiene': 0}	{'longitude': 0.0924199968576431, 'latitude': ...
3	64b469d9fb4578cd17f9383a	1069652	TIWA N TIWA African Restaurant Ltd	Restaurant/Cafe/Canteen	5	{'Hygiene': 5}	{'longitude': 0.0927429, 'latitude': 51.4870351}
4	64b469d9fb4578cd17f93871	1380578	Howe and Co Fish and Chips - Van 17	Mobile caterer	5	{'Hygiene': 0}	{'longitude': 0.0925370007753372, 'latitude': ...

In [28]:
```
# Convert the result to a Pandas DataFrame
df = pd.DataFrame(data)

# Display the number of rows in the DataFrame
print(df.shape)

# Display the first 10 rows of the DataFrame
df.head(10)
```

```
(56, 2)
```

Out[28]:

	_id	num_items
0	Thanet	1130
1	Greenwich	882
2	Maidstone	713
3	Newham	711
4	Swale	686
5	Chelmsford	680
6	Medway	672
7	Bexley	607
8	Southend-On-Sea	586
9	Tendring	542

In [29]:
```
df.sort_values(by="num_items").tail(10).plot(kind="barh", y="num_items", x="_id")
plt.show()
```