GROUP - E

Tushar Raj

Hinal Gala

Ahalya Macharla

Angad Partap Singh

1. Print the first 10 lines of the JSON file

We will start by importing our dataset named "restaurants.json" file using the `open()` function and iterates over the first 10 lines of the file.

In [1]:

```
# Import the json module
import json

with open("restaurants.json", "r") as f:
    for line in list(f)[:10]:
        print(line)
```

```
[{"address": {"building": "1007", "coord": [-73.856077, 40.848447], "street":
"Morris Park Ave", "zipcode": "10462"}, "borough": "Bronx", "cuisine": "Baker
y", "grades": [{"date": {"$date": 1393804800000}, "grade": "A", "score": 2},
{"date": {"$date": 1378857600000}, "grade": "A", "score": 6}, {"date": {"$dat
e": 1358985600000}, "grade": "A", "score": 10}, {"date": {"$date": 1322006400
000}, "grade": "A", "score": 9}, {"date": {"$date": 1299715200000}, "grade":
"B", "score": 14}], "name": "Morris Park Bake Shop", "restaurant_id": "300754
45"}

,{"address": {"building": "469", "coord": [-73.961704, 40.662942], "street":
"Flatbush Avenue", "zipcode": "11225"}, "borough": "Brooklyn", "cuisine": "Ha
mburgers", "grades": [{"date": {"$date": 1419897600000}, "grade": "A", "scor
e": 8}, {"date": {"$date": 1404172800000}, "grade": "B", "score": 23}, {"dat
e": {"$date": 1367280000000}, "grade": "A", "score": 12}, {"date": {"$date":
1336435200000}, "grade": "A", "score": 12}], "name": "Wendy'S", "restaurant_i
d": "30112340"}

,{"address": {"building": "351", "coord": [-73.98513559999999, 40.7676919],
"street": "West   57 Street", "zipcode": "10019"}, "borough": "Manhattan", "c
uisine": "Irish", "grades": [{"date": {"$date": 1409961600000}, "grade": "A",
"score": 2}, {"date": {"$date": 1374451200000}, "grade": "A", "score": 11},
{"date": {"$date": 1343692800000}, "grade": "A", "score": 12}, {"date": {"$da
te": 1325116800000}, "grade": "A", "score": 12}], "name": "Dj Reynolds Pub An
d Restaurant", "restaurant_id": "30191841"}

,{"address": {"building": "2780", "coord": [-73.98241999999999, 40.579505],
"street": "Stillwell Avenue", "zipcode": "11224"}, "borough": "Brooklyn", "cu
isine": "American ", "grades": [{"date": {"$date": 1402358400000}, "grade":
"A", "score": 5}, {"date": {"$date": 1370390400000}, "grade": "A", "score":
7}, {"date": {"$date": 1334275200000}, "grade": "A", "score": 12}, {"date":
{"$date": 1318377600000}, "grade": "A", "score": 12}], "name": "Riviera Cater
er", "restaurant_id": "40356018"}

,{"address": {"building": "97-22", "coord": [-73.8601152, 40.7311739], "stree
t": "63 Road", "zipcode": "11374"}, "borough": "Queens", "cuisine": "Jewish/K
osher", "grades": [{"date": {"$date": 1416787200000}, "grade": "Z", "score":
20}, {"date": {"$date": 1358380800000}, "grade": "A", "score": 13}, {"date":
{"$date": 1343865600000}, "grade": "A", "score": 13}, {"date": {"$date": 1323
907200000}, "grade": "B", "score": 25}], "name": "Tov Kosher Kitchen", "resta
urant_id": "40356068"}

,{"address": {"building": "8825", "coord": [-73.8803827, 40.7643124], "stree
t": "Astoria Boulevard", "zipcode": "11369"}, "borough": "Queens", "cuisine":
"American ", "grades": [{"date": {"$date": 1416009600000}, "grade": "Z", "sco
re": 38}, {"date": {"$date": 1398988800000}, "grade": "A", "score": 10}, {"da
te": {"$date": 1362182400000}, "grade": "A", "score": 7}, {"date": {"$date":
1328832000000}, "grade": "A", "score": 13}], "name": "Brunos On The Boulevar
d", "restaurant_id": "40356151"}

,{"address": {"building": "2206", "coord": [-74.1377286, 40.6119572], "stree
t": "Victory Boulevard", "zipcode": "10314"}, "borough": "Staten Island", "cu
isine": "Jewish/Kosher", "grades": [{"date": {"$date": 1412553600000}, "grad
e": "A", "score": 9}, {"date": {"$date": 1400544000000}, "grade": "A", "scor
e": 12}, {"date": {"$date": 1365033600000}, "grade": "A", "score": 12}, {"dat
e": {"$date": 1327363200000}, "grade": "A", "score": 9}], "name": "Kosher Isl
and", "restaurant_id": "40356442"}
```

```
,{"address": {"building": "7114", "coord": [-73.9068506, 40.6199034], "stree
t": "Avenue U", "zipcode": "11234"}, "borough": "Brooklyn", "cuisine": "Delic
atessen", "grades": [{"date": {"$date": 1401321600000}, "grade": "A", "scor
e": 10}, {"date": {"$date": 1389657600000}, "grade": "A", "score": 10}, {"dat
e": {"$date": 1375488000000}, "grade": "A", "score": 8}, {"date": {"$date": 1
342569600000}, "grade": "A", "score": 10}, {"date": {"$date": 1331251200000},
"grade": "A", "score": 13}, {"date": {"$date": 1318550400000}, "grade": "A",
"score": 9}], "name": "Wilken'S Fine Food", "restaurant_id": "40356483"}

,{"address": {"building": "6409", "coord": [-74.00528899999999, 40.628886],
"street": "11 Avenue", "zipcode": "11219"}, "borough": "Brooklyn", "cuisine":
"American ", "grades": [{"date": {"$date": 1405641600000}, "grade": "A", "sco
re": 12}, {"date": {"$date": 1375142400000}, "grade": "A", "score": 12}, {"da
te": {"$date": 1360713600000}, "grade": "A", "score": 11}, {"date": {"$date":
1345075200000}, "grade": "A", "score": 2}, {"date": {"$date": 1313539200000},
"grade": "A", "score": 11}], "name": "Regina Caterers", "restaurant_id": "403
56649"}

,{"address": {"building": "1839", "coord": [-73.9482609, 40.6408271], "stree
t": "Nostrand Avenue", "zipcode": "11226"}, "borough": "Brooklyn", "cuisine":
"Ice Cream, Gelato, Yogurt, Ices", "grades": [{"date": {"$date": 140529600000
0}, "grade": "A", "score": 12}, {"date": {"$date": 1373414400000}, "grade":
"A", "score": 8}, {"date": {"$date": 1341964800000}, "grade": "A", "score":
5}, {"date": {"$date": 1329955200000}, "grade": "A", "score": 8}], "name": "T
aste The Tropics Ice Cream", "restaurant_id": "40356731"}
```

2. Connecting to the cluster created in MongoDB Atlas.

We will now connect to a MongoDB Atlas cluster using the connection URI provided, and issue a request to confirm the successful connection.Finally, if the request is successful, print a success message and throw an exception message if an exception happens throughout the process.

In [2]:

```python
from pymongo.mongo_client import MongoClient
from pymongo.server_api import ServerApi

uri = "mongodb+srv://tusharraj2890:rBl0gcFJQDDOTD0n@cluster0.muzstyk.mongodb

# Create a new client and connect to the server
client = MongoClient(uri, server_api=ServerApi('1'))

# Send a ping to confirm a successful connection
try:
    client.admin.command('ping')
    print("Pinged your deployment. You successfully connected to MongoDB!")
except Exception as e:
    print(e)
```

```
Pinged your deployment. You successfully connected to MongoDB!
```

Reading the json file and storing in a variable

The code below will read the contents of the "restaurants.json" file and save them in the json_data variable.

In [3]:

```
with open('restaurants.json', 'r') as f:
    json_data = json.loads(f.read())
```

In [4]:

```
json_data
```

```
Out[4]: [{'address': {'building': '1007',
           'coord': [-73.856077, 40.848447],
           'street': 'Morris Park Ave',
           'zipcode': '10462'},
          'borough': 'Bronx',
          'cuisine': 'Bakery',
          'grades': [{'date': {'$date': 1393804800000}, 'grade': 'A', 'score': 2},
           {'date': {'$date': 1378857600000}, 'grade': 'A', 'score': 6},
           {'date': {'$date': 1358985600000}, 'grade': 'A', 'score': 10},
           {'date': {'$date': 1322006400000}, 'grade': 'A', 'score': 9},
           {'date': {'$date': 1299715200000}, 'grade': 'B', 'score': 14}],
          'name': 'Morris Park Bake Shop',
          'restaurant_id': '30075445'},
         {'address': {'building': '469',
           'coord': [-73.961704, 40.662942],
           'street': 'Flatbush Avenue',
           'zipcode': '11225'},
          'borough': 'Brooklyn',
          'cuisine': 'Hamburgers',
          'grades': [{'date': {'$date': 1419897600000}, 'grade': 'A', 'score': 8},
           {'date': {'$date': 1404172800000}, 'grade': 'B', 'score': 23},
           {'date': {'$date': 1367280000000}, 'grade': 'A', 'score': 12},
           {'date': {'$date': 1336435200000}, 'grade': 'A', 'score': 12}],
          'name': "Wendy'S",
          'restaurant_id': '30112340'},
         {'address': {'building': '351',
           'coord': [-73.98513559999999, 40.7676919],
           'street': 'West   57 Street',
           'zipcode': '10019'},
          'borough': 'Manhattan',
          'cuisine': 'Irish',
          'grades': [{'date': {'$date': 1409961600000}, 'grade': 'A', 'score': 2},
           {'date': {'$date': 1374451200000}, 'grade': 'A', 'score': 11},
           {'date': {'$date': 1343692800000}, 'grade': 'A', 'score': 12},
           {'date': {'$date': 1325116800000}, 'grade': 'A', 'score': 12}],
          'name': 'Dj Reynolds Pub And Restaurant',
          'restaurant_id': '30191841'},
         {'address': {'building': '2780',
           'coord': [-73.98241999999999, 40.579505],
           'street': 'Stillwell Avenue',
           'zipcode': '11224'},
          'borough': 'Brooklyn',
          'cuisine': 'American ',
          'grades': [{'date': {'$date': 1402358400000}, 'grade': 'A', 'score': 5},
           {'date': {'$date': 1370390400000}, 'grade': 'A', 'score': 7},
           {'date': {'$date': 1334275200000}, 'grade': 'A', 'score': 12},
           {'date': {'$date': 1318377600000}, 'grade': 'A', 'score': 12}],
          'name': 'Riviera Caterer',
          'restaurant_id': '40356018'},
         {'address': {'building': '97-22',
           'coord': [-73.8601152, 40.7311739],
           'street': '63 Road',
           'zipcode': '11374'},
          'borough': 'Queens',
          'cuisine': 'Jewish/Kosher',
          'grades': [{'date': {'$date': 1416787200000}, 'grade': 'Z', 'score': 20},
  {'date': {'$date': 1358380800000}, 'grade': 'A', 'score': 13},
  {'date': {'$date': 1343865600000}, 'grade': 'A', 'score': 13},
  {'date': {'$date': 1323907200000}, 'grade': 'B', 'score': 25}],
 'name': 'Tov Kosher Kitchen',
 'restaurant_id': '40356068'},
{'address': {'building': '8825',
  'coord': [-73.8803827, 40.7643124],
  'street': 'Astoria Boulevard',
  'zipcode': '11369'},
 'borough': 'Queens',
 'cuisine': 'American ',
 'grades': [{'date': {'$date': 1416009600000}, 'grade': 'Z', 'score': 38},
  {'date': {'$date': 1398988800000}, 'grade': 'A', 'score': 10},
  {'date': {'$date': 1362182400000}, 'grade': 'A', 'score': 7},
  {'date': {'$date': 1328832000000}, 'grade': 'A', 'score': 13}],
 'name': 'Brunos On The Boulevard',
 'restaurant_id': '40356151'},
{'address': {'building': '2206',
  'coord': [-74.1377286, 40.6119572],
  'street': 'Victory Boulevard',
  'zipcode': '10314'},
 'borough': 'Staten Island',
 'cuisine': 'Jewish/Kosher',
 'grades': [{'date': {'$date': 1412553600000}, 'grade': 'A', 'score': 9},
  {'date': {'$date': 1400544000000}, 'grade': 'A', 'score': 12},
  {'date': {'$date': 1365033600000}, 'grade': 'A', 'score': 12},
  {'date': {'$date': 1327363200000}, 'grade': 'A', 'score': 9}],
 'name': 'Kosher Island',
 'restaurant_id': '40356442'},
{'address': {'building': '7114',
  'coord': [-73.9068506, 40.6199034],
  'street': 'Avenue U',
  'zipcode': '11234'},
 'borough': 'Brooklyn',
 'cuisine': 'Delicatessen',
 'grades': [{'date': {'$date': 1401321600000}, 'grade': 'A', 'score': 10},
  {'date': {'$date': 1389657600000}, 'grade': 'A', 'score': 10},
  {'date': {'$date': 1375488000000}, 'grade': 'A', 'score': 8},
  {'date': {'$date': 1342569600000}, 'grade': 'A', 'score': 10},
  {'date': {'$date': 1331251200000}, 'grade': 'A', 'score': 13},
  {'date': {'$date': 1318550400000}, 'grade': 'A', 'score': 9}],
 'name': "Wilken'S Fine Food",
 'restaurant_id': '40356483'},
{'address': {'building': '6409',
  'coord': [-74.00528899999999, 40.628886],
  'street': '11 Avenue',
  'zipcode': '11219'},
 'borough': 'Brooklyn',
 'cuisine': 'American ',
 'grades': [{'date': {'$date': 1405641600000}, 'grade': 'A', 'score': 12},
  {'date': {'$date': 1375142400000}, 'grade': 'A', 'score': 12},
  {'date': {'$date': 1360713600000}, 'grade': 'A', 'score': 11},
  {'date': {'$date': 1345075200000}, 'grade': 'A', 'score': 2},
  {'date': {'$date': 1313539200000}, 'grade': 'A', 'score': 11}],
 'name': 'Regina Caterers',
 'restaurant_id': '40356649'},
{'address': {'building': '1839',
  'coord': [-73.9482609, 40.6408271],
  'street': 'Nostrand Avenue',
  'zipcode': '11226'},
 'borough': 'Brooklyn',
 'cuisine': 'Ice Cream, Gelato, Yogurt, Ices',
 'grades': [{'date': {'$date': 1405296000000}, 'grade': 'A', 'score': 12},
  {'date': {'$date': 1373414400000}, 'grade': 'A', 'score': 8},
  {'date': {'$date': 1341964800000}, 'grade': 'A', 'score': 5},
  {'date': {'$date': 1329955200000}, 'grade': 'A', 'score': 8}],
 'name': 'Taste The Tropics Ice Cream',
 'restaurant_id': '40356731'},
{'address': {'building': '2300',
  'coord': [-73.8786113, 40.8502883],
  'street': 'Southern Boulevard',
  'zipcode': '10460'},
 'borough': 'Bronx',
 'cuisine': 'American ',
 'grades': [{'date': {'$date': 1401235200000}, 'grade': 'A', 'score': 11},
  {'date': {'$date': 1371600000000}, 'grade': 'A', 'score': 4},
  {'date': {'$date': 1339718400000}, 'grade': 'A', 'score': 3}],
 'name': 'Wild Asia',
 'restaurant_id': '40357217'},
{'address': {'building': '7715',
  'coord': [-73.9973325, 40.61174889999999],
  'street': '18 Avenue',
  'zipcode': '11214'},
 'borough': 'Brooklyn',
 'cuisine': 'American ',
 'grades': [{'date': {'$date': 1397606400000}, 'grade': 'A', 'score': 5},
  {'date': {'$date': 1366675200000}, 'grade': 'A', 'score': 2},
  {'date': {'$date': 1335225600000}, 'grade': 'A', 'score': 5},
  {'date': {'$date': 1323993600000}, 'grade': 'A', 'score': 2}],
 'name': 'C & C Catering Service',
 'restaurant_id': '40357437'},
{'address': {'building': '1269',
  'coord': [-73.871194, 40.6730975],
  'street': 'Sutter Avenue',
  'zipcode': '11208'},
 'borough': 'Brooklyn',
 'cuisine': 'Chinese',
 'grades': [{'date': {'$date': 1410825600000}, 'grade': 'B', 'score': 21},
  {'date': {'$date': 1377648000000}, 'grade': 'A', 'score': 7},
  {'date': {'$date': 1364860800000}, 'grade': 'C', 'score': 56},
  {'date': {'$date': 1344988800000}, 'grade': 'B', 'score': 27},
  {'date': {'$date': 1332892800000}, 'grade': 'B', 'score': 27}],
 'name': 'May May Kitchen',
 'restaurant_id': '40358429'},
{'address': {'building': '1',
  'coord': [-73.96926909999999, 40.7685235],
  'street': 'East   66 Street',
  'zipcode': '10065'},
 'borough': 'Manhattan',
 'cuisine': 'American ',
 'grades': [{'date': {'$date': 1399420800000}, 'grade': 'A', 'score': 3},
  {'date': {'$date': 1367539200000}, 'grade': 'A', 'score': 4},
  {'date': {'$date': 1335744000000}, 'grade': 'A', 'score': 6},
  {'date': {'$date': 1324944000000}, 'grade': 'A', 'score': 0}],
 'name': '1 East 66Th Street Kitchen',
 'restaurant_id': '40359480'},
{'address': {'building': '705',
  'coord': [-73.9653967, 40.6064339],
  'street': 'Kings Highway',
  'zipcode': '11223'},
 'borough': 'Brooklyn',
 'cuisine': 'Jewish/Kosher',
 'grades': [{'date': {'$date': 1415577600000}, 'grade': 'A', 'score': 11},
  {'date': {'$date': 1381363200000}, 'grade': 'A', 'score': 13},
  {'date': {'$date': 1349308800000}, 'grade': 'A', 'score': 7},
  {'date': {'$date': 1337558400000}, 'grade': 'A', 'score': 9},
  {'date': {'$date': 1325203200000}, 'grade': 'B', 'score': 19}],
 'name': 'Seuda Foods',
 'restaurant_id': '40360045'},
{'address': {'building': '203',
  'coord': [-73.97822040000001, 40.6435254],
  'street': 'Church Avenue',
  'zipcode': '11218'},
 'borough': 'Brooklyn',
 'cuisine': 'Ice Cream, Gelato, Yogurt, Ices',
 'grades': [{'date': {'$date': 1391990400000}, 'grade': 'A', 'score': 2},
  {'date': {'$date': 1357084800000}, 'grade': 'A', 'score': 13},
  {'date': {'$date': 1326067200000}, 'grade': 'A', 'score': 3},
  {'date': {'$date': 1320624000000}, 'grade': 'P', 'score': 12},
  {'date': {'$date': 1311206400000}, 'grade': 'A', 'score': 13}],
 'name': 'Carvel Ice Cream',
 'restaurant_id': '40360076'},
{'address': {'building': '265-15',
  'coord': [-73.7032601, 40.7386417],
  'street': 'Hillside Avenue',
  'zipcode': '11004'},
 'borough': 'Queens',
 'cuisine': 'Ice Cream, Gelato, Yogurt, Ices',
 'grades': [{'date': {'$date': 1414454400000}, 'grade': 'A', 'score': 9},
  {'date': {'$date': 1379462400000}, 'grade': 'A', 'score': 10},
  {'date': {'$date': 1348099200000}, 'grade': 'A', 'score': 13}],
 'name': 'Carvel Ice Cream',
 'restaurant_id': '40361322'},
{'address': {'building': '6909',
  'coord': [-74.0259567, 40.6353674],
  'street': '3 Avenue',
  'zipcode': '11209'},
 'borough': 'Brooklyn',
 'cuisine': 'Delicatessen',
 'grades': [{'date': {'$date': 1408579200000}, 'grade': 'A', 'score': 4},
  {'date': {'$date': 1393977600000}, 'grade': 'A', 'score': 3},
  {'date': {'$date': 1357776000000}, 'grade': 'A', 'score': 10}],
 'name': 'Nordic Delicacies',
 'restaurant_id': '40361390'},
{'address': {'building': '522',
  'coord': [-73.95171, 40.767461],
  'street': 'East   74 Street',
  'zipcode': '10021'},
  'borough': 'Manhattan',
  'cuisine': 'American ',
  'grades': [{'date': {'$date': 1409616000000}, 'grade': 'A', 'score': 12},
   {'date': {'$date': 1387411200000}, 'grade': 'B', 'score': 16},
   {'date': {'$date': 1369699200000}, 'grade': 'A', 'score': 9},
   {'date': {'$date': 1354838400000}, 'grade': 'A', 'score': 13},
   {'date': {'$date': 1332979200000}, 'grade': 'A', 'score': 11}],
  'name': 'Glorious Food',
  'restaurant_id': '40361521'},
 {'address': {'building': '284',
   'coord': [-73.9829239, 40.6580753],
   'street': 'Prospect Park West',
   'zipcode': '11215'},
  'borough': 'Brooklyn',
  'cuisine': 'American ',
  'grades': [{'date': {'$date': 1416355200000}, 'grade': 'A', 'score': 11},
   {'date': {'$date': 1384387200000}, 'grade': 'A', 'score': 2},
   {'date': {'$date': 1354665600000}, 'grade': 'A', 'score': 13},
   {'date': {'$date': 1337212800000}, 'grade': 'A', 'score': 11}],
  'name': 'The Movable Feast',
  'restaurant_id': '40361606'},
 {'address': {'building': '129-08',
   'coord': [-73.839297, 40.78147],
   'street': '20 Avenue',
   'zipcode': '11356'},
  'borough': 'Queens',
  'cuisine': 'Delicatessen',
  'grades': [{'date': {'$date': 1408147200000}, 'grade': 'A', 'score': 12},
   {'date': {'$date': 1377561600000}, 'grade': 'A', 'score': 9},
   {'date': {'$date': 1348099200000}, 'grade': 'A', 'score': 7},
   {'date': {'$date': 1317254400000}, 'grade': 'A', 'score': 10}],
  'name': "Sal'S Deli",
  'restaurant_id': '40361618'},
 {'address': {'building': '759',
   'coord': [-73.9925306, 40.7309346],
   'street': 'Broadway',
   'zipcode': '10003'},
  'borough': 'Manhattan',
  'cuisine': 'Delicatessen',
  'grades': [{'date': {'$date': 1390262400000}, 'grade': 'A', 'score': 12},
   {'date': {'$date': 1357257600000}, 'grade': 'A', 'score': 11},
   {'date': {'$date': 1339027200000}, 'grade': 'A', 'score': 6},
   {'date': {'$date': 1326758400000}, 'grade': 'A', 'score': 8}],
  'name': "Bully'S Deli",
  'restaurant_id': '40361708'},
 {'address': {'building': '3406',
   'coord': [-73.94024739999999, 40.7623288],
   'street': '10 Street',
   'zipcode': '11106'},
  'borough': 'Queens',
  'cuisine': 'Delicatessen',
  'grades': [{'date': {'$date': 1395187200000}, 'grade': 'A', 'score': 3},
   {'date': {'$date': 1363132800000}, 'grade': 'A', 'score': 12},
   {'date': {'$date': 1332806400000}, 'grade': 'A', 'score': 8},
   {'date': {'$date': 1301961600000}, 'grade': 'A', 'score': 7}],
  'name': "Steve Chu'S Deli & Grocery",
  'restaurant_id': '40361998'},
 {'address': {'building': '502',
   'coord': [-73.976112, 40.786714],
   'street': 'Amsterdam Avenue',
   'zipcode': '10024'},
  'borough': 'Manhattan',
  'cuisine': 'Chicken',
  'grades': [{'date': {'$date': 1410739200000}, 'grade': 'A', 'score': 10},
   {'date': {'$date': 1393891200000}, 'grade': 'A', 'score': 13},
   {'date': {'$date': 1374105600000}, 'grade': 'A', 'score': 13},
   {'date': {'$date': 1357689600000}, 'grade': 'A', 'score': 11},
   {'date': {'$date': 1334016000000}, 'grade': 'A', 'score': 10},
   {'date': {'$date': 1321315200000}, 'grade': 'A', 'score': 7}],
  'name': "Harriet'S Kitchen",
  'restaurant_id': '40362098'},
 {'address': {'building': '730',
   'coord': [-73.96805719999999, 40.7925587],
   'street': 'Columbus Avenue',
   'zipcode': '10025'},
  'borough': 'Manhattan',
  'cuisine': 'American ',
  'grades': [{'date': {'$date': 1410480000000}, 'grade': 'B', 'score': 26},
   {'date': {'$date': 1377648000000}, 'grade': 'A', 'score': 9},
   {'date': {'$date': 1364169600000}, 'grade': 'B', 'score': 20},
   {'date': {'$date': 1329177600000}, 'grade': 'A', 'score': 12}],
  'name': 'P & S Deli Grocery',
  'restaurant_id': '40362264'},
 {'address': {'building': '18',
   'coord': [-73.996984, 40.72589],
   'street': 'West Houston Street',
   'zipcode': '10012'},
  'borough': 'Manhattan',
  'cuisine': 'American ',
  'grades': [{'date': {'$date': 1396483200000}, 'grade': 'A', 'score': 9},
   {'date': {'$date': 1365120000000}, 'grade': 'A', 'score': 4},
   {'date': {'$date': 1332288000000}, 'grade': 'A', 'score': 13},
   {'date': {'$date': 1303862400000}, 'grade': 'A', 'score': 5}],
  'name': 'Angelika Film Center',
  'restaurant_id': '40362274'},
 {'address': {'building': '531',
   'coord': [-73.9634876, 40.6940001],
   'street': 'Myrtle Avenue',
   'zipcode': '11205'},
  'borough': 'Brooklyn',
  'cuisine': 'Hamburgers',
  'grades': [{'date': {'$date': 1395100800000}, 'grade': 'A', 'score': 8},
   {'date': {'$date': 1363564800000}, 'grade': 'A', 'score': 8},
   {'date': {'$date': 1349827200000}, 'grade': 'A', 'score': 7},
   {'date': {'$date': 1316649600000}, 'grade': 'A', 'score': 2}],
  'name': 'White Castle',
  'restaurant_id': '40362344'},
 {'address': {'building': '103-05',
   'coord': [-73.8642349, 40.75356],
   'street': '37 Avenue',
   'zipcode': '11368'},
  'borough': 'Queens',
   'cuisine': 'Chinese',
   'grades': [{'date': {'$date': 1398038400000}, 'grade': 'A', 'score': 10},
    {'date': {'$date': 1384214400000}, 'grade': 'A', 'score': 5},
    {'date': {'$date': 1370304000000}, 'grade': 'A', 'score': 12},
    {'date': {'$date': 1352851200000}, 'grade': 'A', 'score': 10},
    {'date': {'$date': 1349913600000}, 'grade': 'P', 'score': 0},
    {'date': {'$date': 1337817600000}, 'grade': 'A', 'score': 13},
    {'date': {'$date': 1323302400000}, 'grade': 'A', 'score': 12},
    {'date': {'$date': 1311120000000}, 'grade': 'A', 'score': 11}],
   'name': 'Ho Mei Restaurant',
   'restaurant_id': '40362432'},
  {'address': {'building': '60',
    'coord': [-74.0085357, 40.70620539999999],
    'street': 'Wall Street',
    'zipcode': '10005'},
   'borough': 'Manhattan',
   'cuisine': 'Turkish',
   'grades': [{'date': {'$date': 1411689600000}, 'grade': 'A', 'score': 9},
    {'date': {'$date': 1379462400000}, 'grade': 'A', 'score': 13},
    {'date': {'$date': 1348185600000}, 'grade': 'A', 'score': 9},
    {'date': {'$date': 1336521600000}, 'grade': 'A', 'score': 11}],
   'name': 'The Country Cafe',
   'restaurant_id': '40362715'},
  {'address': {'building': '195',
    'coord': [-73.9246028, 40.6522396],
    'street': 'East   56 Street',
    'zipcode': '11203'},
   'borough': 'Brooklyn',
   'cuisine': 'Caribbean',
   'grades': [{'date': {'$date': 1399939200000}, 'grade': 'A', 'score': 2},
    {'date': {'$date': 1367971200000}, 'grade': 'A', 'score': 7},
    {'date': {'$date': 1348272000000}, 'grade': 'A', 'score': 11},
    {'date': {'$date': 1307318400000}, 'grade': 'A', 'score': 12}],
   'name': "Shashemene Int'L Restaura",
   'restaurant_id': '40362869'},
  {'address': {'building': '107',
    'coord': [-74.00920839999999, 40.7132925],
    'street': 'Church Street',
    'zipcode': '10007'},
   'borough': 'Manhattan',
   'cuisine': 'American ',
   'grades': [{'date': {'$date': 1405641600000}, 'grade': 'A', 'score': 12},
    {'date': {'$date': 1393372800000}, 'grade': 'A', 'score': 9},
    {'date': {'$date': 1377475200000}, 'grade': 'A', 'score': 9},
    {'date': {'$date': 1359676800000}, 'grade': 'A', 'score': 12},
    {'date': {'$date': 1326758400000}, 'grade': 'A', 'score': 13},
    {'date': {'$date': 1318896000000}, 'grade': 'A', 'score': 11}],
   'name': 'Downtown Deli',
   'restaurant_id': '40363021'},
  {'address': {'building': '1006',
    'coord': [-73.84856870000002, 40.8903781],
    'street': 'East 233 Street',
    'zipcode': '10466'},
   'borough': 'Bronx',
   'cuisine': 'Ice Cream, Gelato, Yogurt, Ices',
   'grades': [{'date': {'$date': 1398297600000}, 'grade': 'A', 'score': 10},
     {'date': {'$date': 1378339200000}, 'grade': 'A', 'score': 10},
     {'date': {'$date': 1361404800000}, 'grade': 'A', 'score': 9},
     {'date': {'$date': 1341273600000}, 'grade': 'A', 'score': 11},
     {'date': {'$date': 1310342400000}, 'grade': 'A', 'score': 5}],
    'name': 'Carvel Ice Cream',
    'restaurant_id': '40363093'},
   {'address': {'building': '56',
     'coord': [-73.991495, 40.692273],
     'street': 'Court Street',
     'zipcode': '11201'},
    'borough': 'Brooklyn',
    'cuisine': 'Donuts',
    'grades': [{'date': {'$date': 1419897600000}, 'grade': 'A', 'score': 8},
     {'date': {'$date': 1389744000000}, 'grade': 'A', 'score': 9},
     {'date': {'$date': 1357603200000}, 'grade': 'A', 'score': 11},
     {'date': {'$date': 1326931200000}, 'grade': 'A', 'score': 10}],
    'name': "Dunkin' Donuts",
    'restaurant_id': '40363098'},
   {'address': {'building': '7615',
     'coord': [-74.0228449, 40.6281815],
     'street': '5 Avenue',
     'zipcode': '11209'},
    'borough': 'Brooklyn',
    'cuisine': 'American ',
    'grades': [{'date': {'$date': 1417651200000}, 'grade': 'A', 'score': 10},
     {'date': {'$date': 1382572800000}, 'grade': 'A', 'score': 11},
     {'date': {'$date': 1366243200000}, 'grade': 'A', 'score': 5},
     {'date': {'$date': 1333584000000}, 'grade': 'A', 'score': 13}],
    'name': 'Mejlander & Mulgannon',
    'restaurant_id': '40363117'}]
```

In [5]:

```
# accessing the database
db = client['Restaurants'];

# accessing the collection
collection = db['Justeat'];
```

Inserting json records in Justeat collection

In [6]:

```
collection.insert_many(json_data)
```

Total number of records present in the collection

In [7]:

```
len(list(collection.find()))
```

Out[7]:

```
34
```

3. Query to print out the name of the Restaurant that has Cuisine = "Chicken" and ZipCode = "10024"

-

The following code searches the MongoDB collection for documents with the 'cuisine' field set to 'Chicken' and the 'address.zipcode' field set to '10024'. It returns the names of the restaurant while excluding the _id field from the result. The query result is then turned into a Python list of dictionaries holding the names of the matched restaurant.

```
In [8]: list(collection.find({'cuisine': 'Chicken', 'address.zipcode': '10024'}, {'r
```

```
Out[8]: [{'name': "Harriet'S Kitchen"}]
```

4. Query to calculate for each borough, the total number of "A" grades.

The code searches for top-grade documents ('A' grade), groups them by borough, counts the number of top-grade documents in each borough, and returns the results as a list of dictionaries.

```
In [9]: list(collection.aggregate([
            {'$unwind': '$grades'},
            {'$match': {'grades.grade': 'A'}},
            {'$group': {
                    "_id": "$borough",
                    "count": {"$sum": 1}
                }
            }
        ]))
```

```
Out[9]: [{'_id': 'Manhattan', 'count': 38},
         {'_id': 'Brooklyn', 'count': 58},
         {'_id': 'Staten Island', 'count': 4},
         {'_id': 'Queens', 'count': 23},
         {'_id': 'Bronx', 'count': 12}]
```

5. Map-Reduce Job

To count how many cuisines types are within each borough.

5.1 Mapper

Below mapper code is broken into two parts:-

1. querying the data from the collection.
2. performs the mapping operation.

For each document in the collection, it extracts the 'borough' and 'cuisine' fields and builds a new dictionary with these values. It also provides a fixed 'count' value of 1 to represent each document as a single occurrence. The result of this dictionary is being appended to the mapper_result variable. The mapper_result list will contain dictionaries representing the 'borough,' 'cuisine,' and a count of 1 for each document in the collection.

In [27]:

```
# querying the data from the collection
documentList = list(collection.find());

# mapper_result -> where mapper result will be stored
mapper_result = [];
for document in documentList:
    mapper_result.append({
        'borough': document["borough"],
        'cuisine': document["cuisine"],
        'count': 1
    });
```

5.2 Sorting

In [28]:

```
#Sorting the data on Borough and cuisine fields
sorted_result = sorted(mapper_result, key = lambda k: (k['borough'], k['cuis
```

In [29]:

```
# print the sorted result
sorted_result
```

```
Out[29]: [{'borough': 'Bronx', 'cuisine': 'American ', 'count': 1},
          {'borough': 'Bronx', 'cuisine': 'Bakery', 'count': 1},
          {'borough': 'Bronx',
           'cuisine': 'Ice Cream, Gelato, Yogurt, Ices',
           'count': 1},
          {'borough': 'Brooklyn', 'cuisine': 'American ', 'count': 1},
          {'borough': 'Brooklyn', 'cuisine': 'American ', 'count': 1},
          {'borough': 'Brooklyn', 'cuisine': 'American ', 'count': 1},
          {'borough': 'Brooklyn', 'cuisine': 'American ', 'count': 1},
          {'borough': 'Brooklyn', 'cuisine': 'American ', 'count': 1},
          {'borough': 'Brooklyn', 'cuisine': 'Caribbean', 'count': 1},
          {'borough': 'Brooklyn', 'cuisine': 'Chinese', 'count': 1},
          {'borough': 'Brooklyn', 'cuisine': 'Delicatessen', 'count': 1},
          {'borough': 'Brooklyn', 'cuisine': 'Delicatessen', 'count': 1},
          {'borough': 'Brooklyn', 'cuisine': 'Donuts', 'count': 1},
          {'borough': 'Brooklyn', 'cuisine': 'Hamburgers', 'count': 1},
          {'borough': 'Brooklyn', 'cuisine': 'Hamburgers', 'count': 1},
          {'borough': 'Brooklyn',
           'cuisine': 'Ice Cream, Gelato, Yogurt, Ices',
           'count': 1},
          {'borough': 'Brooklyn',
           'cuisine': 'Ice Cream, Gelato, Yogurt, Ices',
           'count': 1},
          {'borough': 'Brooklyn', 'cuisine': 'Jewish/Kosher', 'count': 1},
          {'borough': 'Manhattan', 'cuisine': 'American ', 'count': 1},
          {'borough': 'Manhattan', 'cuisine': 'American ', 'count': 1},
          {'borough': 'Manhattan', 'cuisine': 'American ', 'count': 1},
          {'borough': 'Manhattan', 'cuisine': 'American ', 'count': 1},
          {'borough': 'Manhattan', 'cuisine': 'American ', 'count': 1},
          {'borough': 'Manhattan', 'cuisine': 'Chicken', 'count': 1},
          {'borough': 'Manhattan', 'cuisine': 'Delicatessen', 'count': 1},
          {'borough': 'Manhattan', 'cuisine': 'Irish', 'count': 1},
          {'borough': 'Manhattan', 'cuisine': 'Turkish', 'count': 1},
          {'borough': 'Queens', 'cuisine': 'American ', 'count': 1},
          {'borough': 'Queens', 'cuisine': 'Chinese', 'count': 1},
          {'borough': 'Queens', 'cuisine': 'Delicatessen', 'count': 1},
          {'borough': 'Queens', 'cuisine': 'Delicatessen', 'count': 1},
          {'borough': 'Queens',
           'cuisine': 'Ice Cream, Gelato, Yogurt, Ices',
           'count': 1},
          {'borough': 'Queens', 'cuisine': 'Jewish/Kosher', 'count': 1},
          {'borough': 'Staten Island', 'cuisine': 'Jewish/Kosher', 'count': 1}]
```

5.3 Reducer

Below code takes sorted data and reduces it to group entries by 'borough' and 'cuisine,' summing the counts for each combination. The result is a string list, with each string representing a borough, cuisine, and count combination.

```
In [60]: # Initialize variables
         result = []
         current_borough = None
         current_cuisine = None
```

```python
count = 0

# Iterate over the sorted data
for entry in sorted_result:
    borough = entry['borough']
    cuisine = entry['cuisine']
    entry_count = entry['count']

    # If it's a new borough and cuisine combination
    if (borough != current_borough) or (cuisine != current_cuisine):
        # Append the previous combination's result (if exists)
        if current_borough is not None:
            result.append(f'{current_borough} {current_cuisine} {count}')

        # Reset the count and update the current combination
        count = entry_count
        current_borough = borough
        current_cuisine = cuisine
    else:
        # Add to the count for the current combination
        count += entry_count

# Append the last combination's result
if current_borough is not None:
    result.append(f'{current_borough} {current_cuisine} {count}')

# Print the reducer result
print('\n'.join(result));
```

```
Bronx American  1
Bronx Bakery 1
Bronx Ice Cream, Gelato, Yogurt, Ices 1
Brooklyn American  5
Brooklyn Caribbean 1
Brooklyn Chinese 1
Brooklyn Delicatessen 2
Brooklyn Donuts 1
Brooklyn Hamburgers 2
Brooklyn Ice Cream, Gelato, Yogurt, Ices 2
Brooklyn Jewish/Kosher 1
Manhattan American  5
Manhattan Chicken 1
Manhattan Delicatessen 1
Manhattan Irish 1
Manhattan Turkish 1
Queens American  1
Queens Chinese 1
Queens Delicatessen 2
Queens Ice Cream, Gelato, Yogurt, Ices 1
Queens Jewish/Kosher 1
Staten Island Jewish/Kosher 1
```