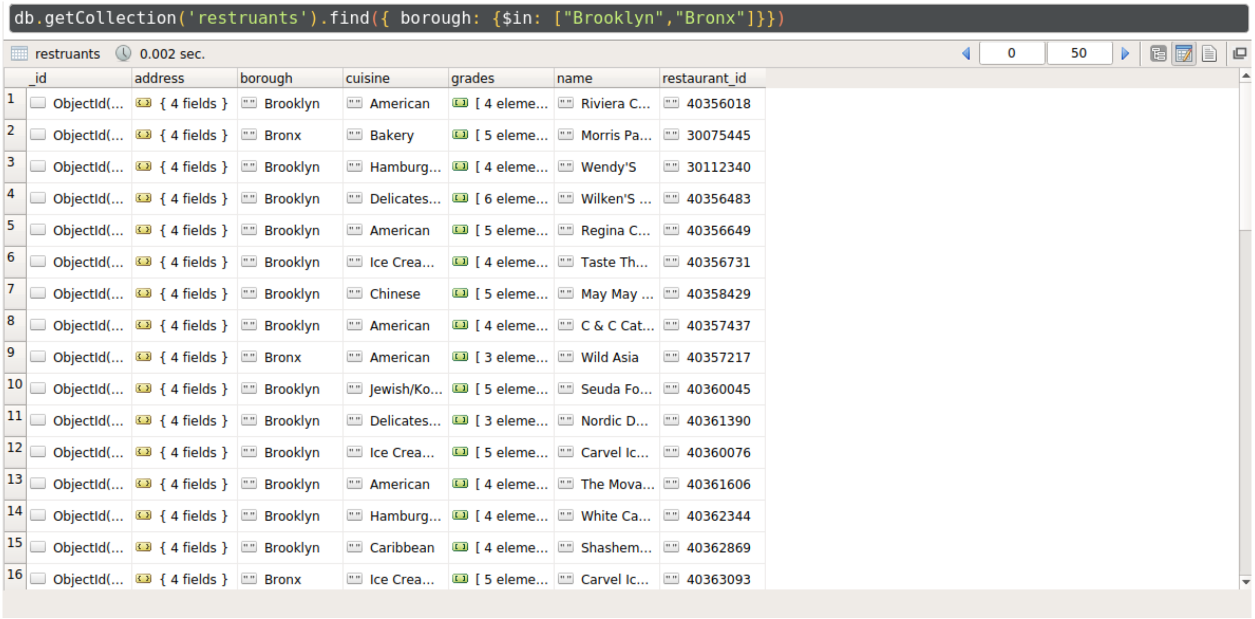
1. MONGODB
Download the file restaurants.json from Canvas and load it into a MongoDB database. The
file contains 3772 documents.
Write the following MongoDB queries on the restaurant's collection (you can write them either directly in the MongoDB shell or inside a python script):

1. Display all the restaurants located in the boroughs Bronx or Brooklyn.
db.getCollection('restruants').find({ borough: {$in: ["Brooklyn","Bronx"]}})



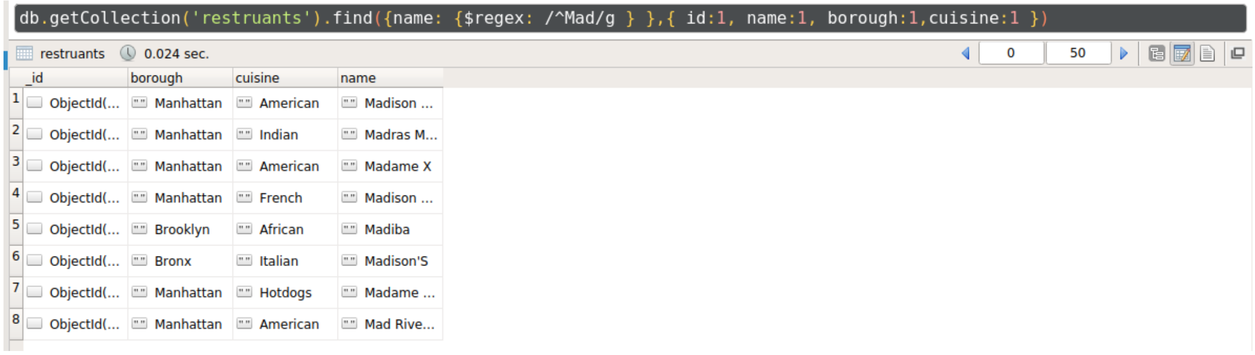
```
db.getCollection('restruants').find({ borough: {$in: ["Brooklyn","Bronx"]}})
```

restruants 0.002 sec.

	_id	address	borough	cuisine	grades	name	restaurant_id
1	ObjectId(...	{ 4 fields }	Brooklyn	American	[4 eleme...	Riviera C...	40356018
2	ObjectId(...	{ 4 fields }	Bronx	Bakery	[5 eleme...	Morris Pa...	30075445
3	ObjectId(...	{ 4 fields }	Brooklyn	Hamburg...	[4 eleme...	Wendy'S	30112340
4	ObjectId(...	{ 4 fields }	Brooklyn	Delicates...	[6 eleme...	Wilken'S ...	40356483
5	ObjectId(...	{ 4 fields }	Brooklyn	American	[5 eleme...	Regina C...	40356649
6	ObjectId(...	{ 4 fields }	Brooklyn	Ice Crea...	[4 eleme...	Taste Th...	40356731
7	ObjectId(...	{ 4 fields }	Brooklyn	Chinese	[5 eleme...	May May ...	40358429
8	ObjectId(...	{ 4 fields }	Brooklyn	American	[4 eleme...	C & C Cat...	40357437
9	ObjectId(...	{ 4 fields }	Bronx	American	[3 eleme...	Wild Asia	40357217
10	ObjectId(...	{ 4 fields }	Brooklyn	Jewish/Ko...	[5 eleme...	Seuda Fo...	40360045
11	ObjectId(...	{ 4 fields }	Brooklyn	Delicates...	[3 eleme...	Nordic D...	40361390
12	ObjectId(...	{ 4 fields }	Brooklyn	Ice Crea...	[5 eleme...	Carvel Ic...	40360076
13	ObjectId(...	{ 4 fields }	Brooklyn	American	[4 eleme...	The Mova...	40361606
14	ObjectId(...	{ 4 fields }	Brooklyn	Hamburg...	[4 eleme...	White Ca...	40362344
15	ObjectId(...	{ 4 fields }	Brooklyn	Caribbean	[4 eleme...	Shashem...	40362869
16	ObjectId(...	{ 4 fields }	Bronx	Ice Crea...	[5 eleme...	Carvel Ic...	40363093

2. Find the restaurant id, name, borough, and cuisine for those restaurants whose name starts with the letters 'Mad'.

db.getCollection('restruants').find({name: {$regex: /^Mad/g } },{ id:1, name:1, borough:1,cuisine:1 })



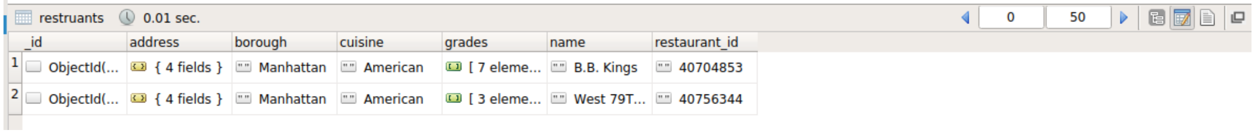
```
db.getCollection('restruants').find({name: {$regex: /^Mad/g } },{ id:1, name:1, borough:1,cuisine:1 })
```

restruants 0.024 sec.

	_id	borough	cuisine	name
1	ObjectId(...	Manhattan	American	Madison ...
2	ObjectId(...	Manhattan	Indian	Madras M...
3	ObjectId(...	Manhattan	American	Madame X
4	ObjectId(...	Manhattan	French	Madison ...
5	ObjectId(...	Brooklyn	African	Madiba
6	ObjectId(...	Bronx	Italian	Madison'S
7	ObjectId(...	Manhattan	Hotdogs	Madame ...
8	ObjectId(...	Manhattan	American	Mad Rive...

3. Find the restaurants that have received a score between 80 and 90.

```
db.getCollection('restruants').find({grades:{$elemMatch: {score:{ $gte:80,$lte:90 }}}})
```

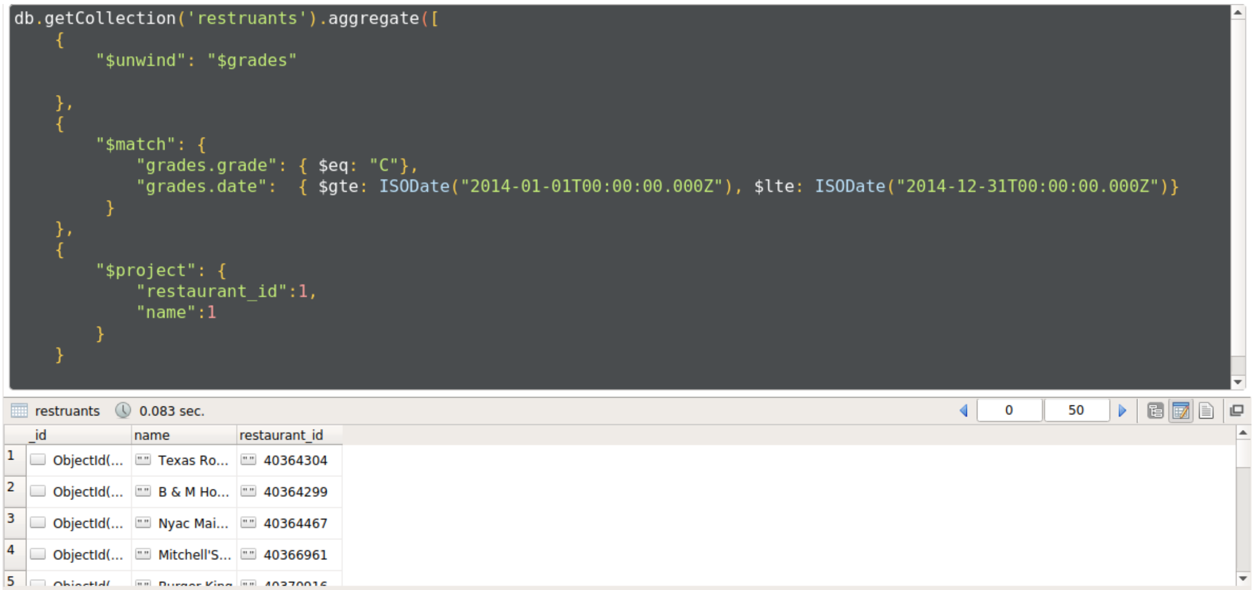


restruants 0.01 sec.

	_id	address	borough	cuisine	grades	name	restaurant_id
1	ObjectId(...	{ 4 fields }	Manhattan	American	[7 eleme...	B.B. Kings	40704853
2	ObjectId(...	{ 4 fields }	Manhattan	American	[3 eleme...	West 79T...	40756344

4. Display the restaurant id and name of restaurants which have received 'C' grade in year
2014.

```
db.getCollection('restruants').aggregate([
   {
      "$unwind": "$grades"
   },
   {
      "$match": {
         "grades.grade": { $eq: "C"},
         "grades.date":  { $gte: ISODate("2014-01-01T00:00:00.000Z"), $lte:
ISODate("2014-12-31T00:00:00.000Z")}
       }
   },
   {
      "$project": {
         "restaurant_id":1,
         "name":1
      }
   }
   ])
```

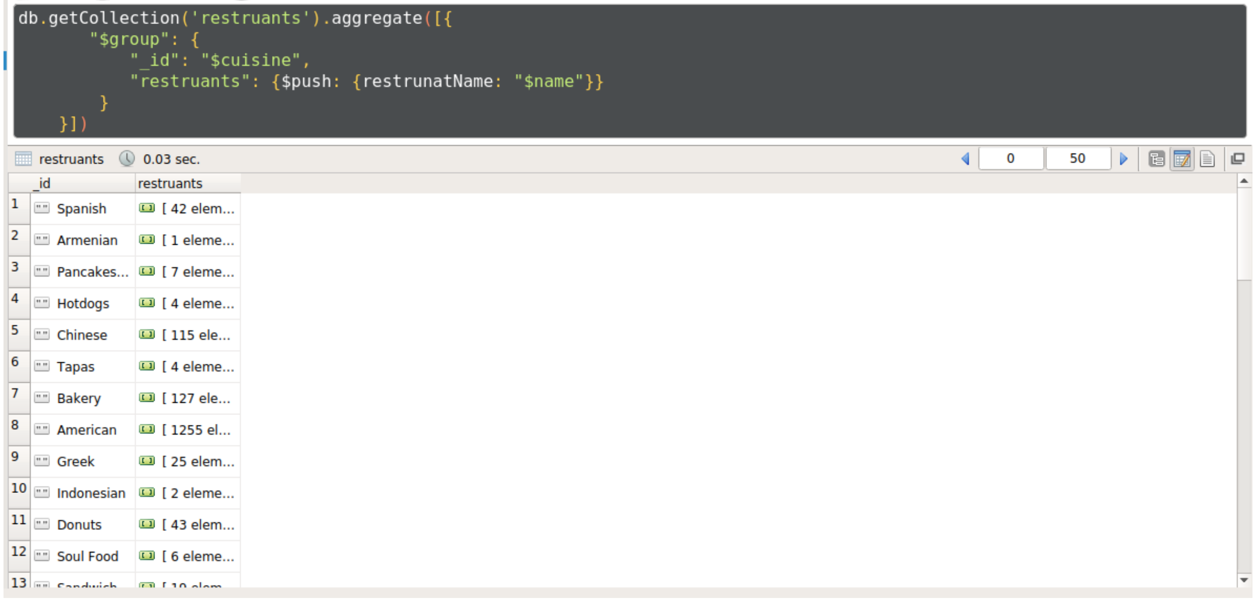


```
db.getCollection('restruants').aggregate([
    {
        "$unwind": "$grades"

    },
    {
        "$match": {
            "grades.grade": { $eq: "C"},
            "grades.date":  { $gte: ISODate("2014-01-01T00:00:00.000Z"), $lte: ISODate("2014-12-31T00:00:00.000Z")}
         }
    },
    {
        "$project": {
            "restaurant_id":1,
            "name":1
        }
    }
```

	_id	name	restaurant_id
1	ObjectId(...	Texas Ro...	40364304
2	ObjectId(...	B & M Ho...	40364299
3	ObjectId(...	Nyac Mai...	40364467
4	ObjectId(...	Mitchell'S...	40366961
5	ObjectId(	Burger King	40370016

5. Find how many restaurants belong to each cuisine (note that the cuisine attribute may contain more than one cuisine type, e.g., ”Ice Cream, Gelato, Yogurt, Ices”).

```
db.getCollection('restruants').aggregate([{
     "$group": {
        "_id": "$cuisine",
        "restruants": {$push: {restrunatName: "$name"}}
      }
  }])
```



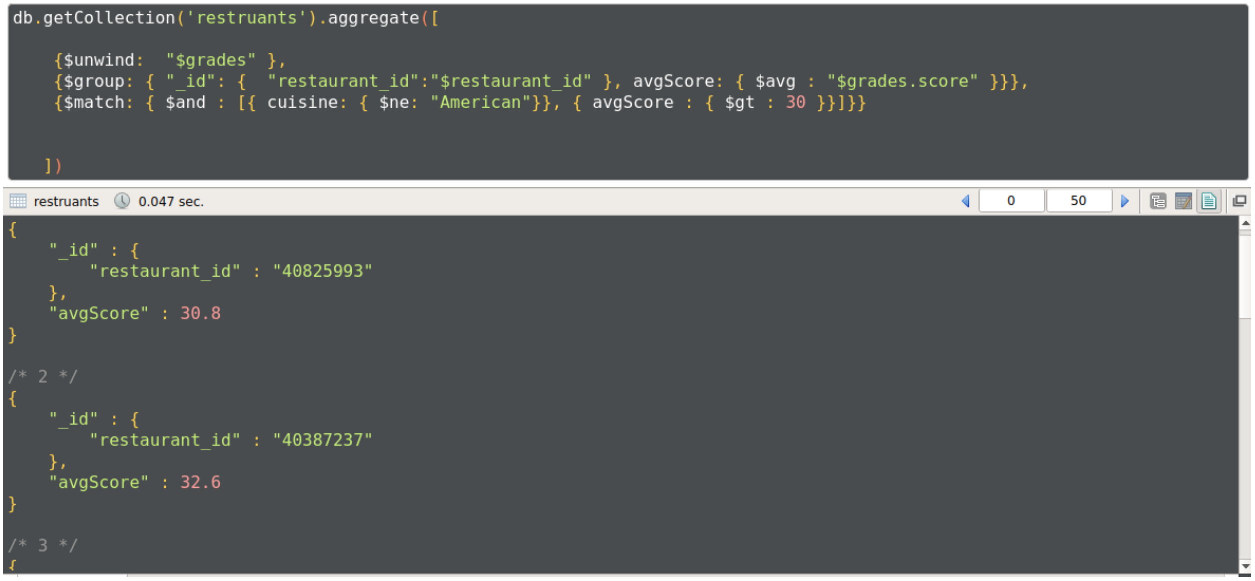
```
db.getCollection('restruants').aggregate([{
        "$group": {
            "_id": "$cuisine",
            "restruants": {$push: {restrunatName: "$name"}}
        }
    }])
```

restruants 0.03 sec.

	_id	restruants
1	Spanish	[42 elem...
2	Armenian	[1 eleme...
3	Pancakes...	[7 eleme...
4	Hotdogs	[4 eleme...
5	Chinese	[115 ele...
6	Tapas	[4 eleme...
7	Bakery	[127 ele...
8	American	[1255 el...
9	Greek	[25 elem...
10	Indonesian	[2 eleme...
11	Donuts	[43 elem...
12	Soul Food	[6 eleme...
13	Sandwich	[10 elem

6. Find the restaurants that do not prepare any cuisine of ’American’ and their average grade score is higher than 30. Display the restaurant id and their average score.

db.getCollection('restruants').aggregate([
   {$unwind: "$grades" },
   {$group: { "_id": { "restaurant_id":"$restaurant_id" }, avgScore: { $avg : "$grades.score" }}},
   {$match: { $and : [{ cuisine: { $ne: "American"}}, { avgScore : { $gt : 30 }}]}}
  ])



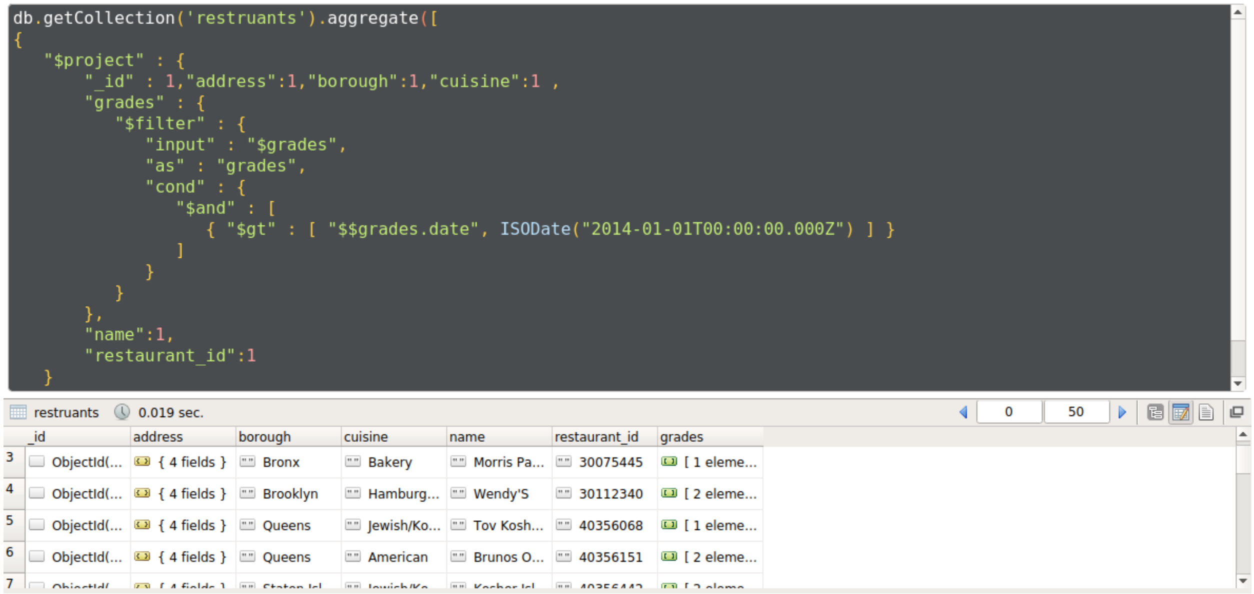
```
db.getCollection('restruants').aggregate([

    {$unwind:  "$grades" },
    {$group: { "_id": {  "restaurant_id":"$restaurant_id" }, avgScore: { $avg : "$grades.score" }}},
    {$match: { $and : [{ cuisine: { $ne: "American"}}, { avgScore : { $gt : 30 }}]}}


   ])
```

restruants 0.047 sec.

```
{
    "_id" : {
        "restaurant_id" : "40825993"
    },
    "avgScore" : 30.8
}

/* 2 */
{
    "_id" : {
        "restaurant_id" : "40387237"
    },
    "avgScore" : 32.6
}

/* 3 */
{
```

7. For each restaurant display only the grades that were recorded from the year 2014 onwards.

```
db.getCollection('restruants').aggregate([
{
  "$project" : {
     "_id" : 1,"address":1,"borough":1,"cuisine":1 ,
     "grades" : {
       "$filter" : {
         "input" : "$grades",
         "as" : "grades",
         "cond" : {
           "$and" : [
             { "$gt" : [ "$$grades.date", ISODate("2014-01-01T00:00:00.000Z") ] }
           ]
         }
       }
     },
     "name":1,
     "restaurant_id":1
  }}])
```



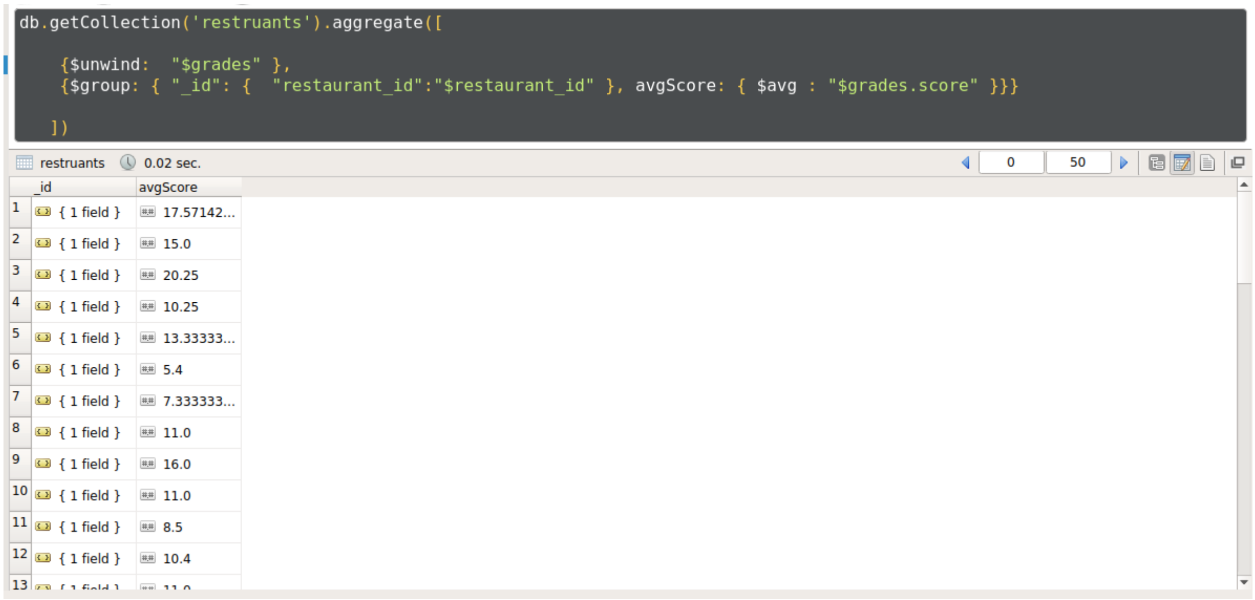
```
db.getCollection('restruants').aggregate([
{
   "$project" : {
       "_id" : 1,"address":1,"borough":1,"cuisine":1 ,
       "grades" : {
          "$filter" : {
             "input" : "$grades",
             "as" : "grades",
             "cond" : {
                "$and" : [
                   { "$gt" : [ "$$grades.date", ISODate("2014-01-01T00:00:00.000Z") ] }
                ]
             }
          }
       },
       "name":1,
       "restaurant_id":1
   }
```

restruants 0.019 sec.

	_id	address	borough	cuisine	name	restaurant_id	grades
3	ObjectId(...	{ 4 fields }	Bronx	Bakery	Morris Pa...	30075445	[1 eleme...
4	ObjectId(...	{ 4 fields }	Brooklyn	Hamburg...	Wendy'S	30112340	[2 eleme...
5	ObjectId(...	{ 4 fields }	Queens	Jewish/Ko...	Tov Kosh...	40356068	[1 eleme...
6	ObjectId(...	{ 4 fields }	Queens	American	Brunos O...	40356151	[2 eleme...
7	ObjectId(...	{ 4 fields }	Staten Isl...	Jewish/Ko...	Kosher Isl...	40356442	[2 eleme...

8. Calculate the average score across all the restaurants in the collection.

```
db.getCollection('restruants').aggregate([

   {$unwind:  "$grades" },
   {$group: { "_id": {  "restaurant_id":"$restaurant_id" }, avgScore: { $avg :
"$grades.score" }}}
  ])
```



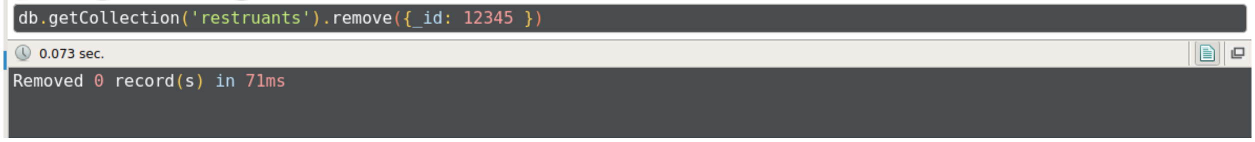
```
db.getCollection('restruants').aggregate([

    {$unwind:  "$grades" },
    {$group: { "_id": {  "restaurant_id":"$restaurant_id" }, avgScore: { $avg : "$grades.score" }}}

   ])
```

restruants 0.02 sec.

0 50

	_id	avgScore
1	{ 1 field }	17.57142...
2	{ 1 field }	15.0
3	{ 1 field }	20.25
4	{ 1 field }	10.25
5	{ 1 field }	13.33333...
6	{ 1 field }	5.4
7	{ 1 field }	7.333333...
8	{ 1 field }	11.0
9	{ 1 field }	16.0
10	{ 1 field }	11.0
11	{ 1 field }	8.5
12	{ 1 field }	10.4
13	{ 1 field }	11.0

9. Remove the restaurant whose id is ’12345’.
db.getCollection('restruants').remove({_id: 12345 })

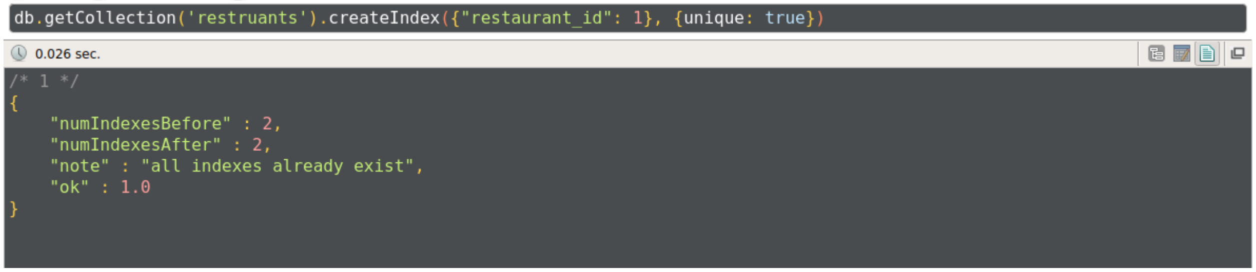


```
db.getCollection('restruants').remove({_id: 12345 })
```

0.073 sec.

```
Removed 0 record(s) in 71ms
```

10. Create index on column restaurant id enforcing uniqueness.
db.getCollection('restruants').createIndex({"restaurant_id": 1}, {unique: true})



```
db.getCollection('restruants').createIndex({"restaurant_id": 1}, {unique: true})
```

0.026 sec.

```
/* 1 */
{
    "numIndexesBefore" : 2,
    "numIndexesAfter" : 2,
    "note" : "all indexes already exist",
    "ok" : 1.0
}
```

2. MapReduce
Describe how to implement the following relational operations using MapReduce. Write the map and reduce functions in pseudocode.

1. Projection π S (R): From each tuple of relation R produce only the components for the attributes in S.

In a projection, the process can work by only utilizing the map portion only but the reduced portions work too
It makes the tuple appear many times

πS(R)

Map - Remove the components that don't have attributes like the ones in S by looping through an element of R, assign the result to a new tuple

- Use the new of the two tuples to create a key-value pair (new tuple, new tuple)

Reduce - For every key in the Map, the tuple fetches the component of the new tuple and returns an array of each of its other occurrences.

2. Intersection R∩S: Return the tuples that are present in both relations R and S. Assume
that relations R and S have the same schema (same attributes and same type).

Here the pseudocode should return a tuple if both relations have tuples

Map - For each tuple x in R or S, return a key and value for the tuple [x, x]

Reduce - Return a key-value pair every time the value list is [x,x] for a key x

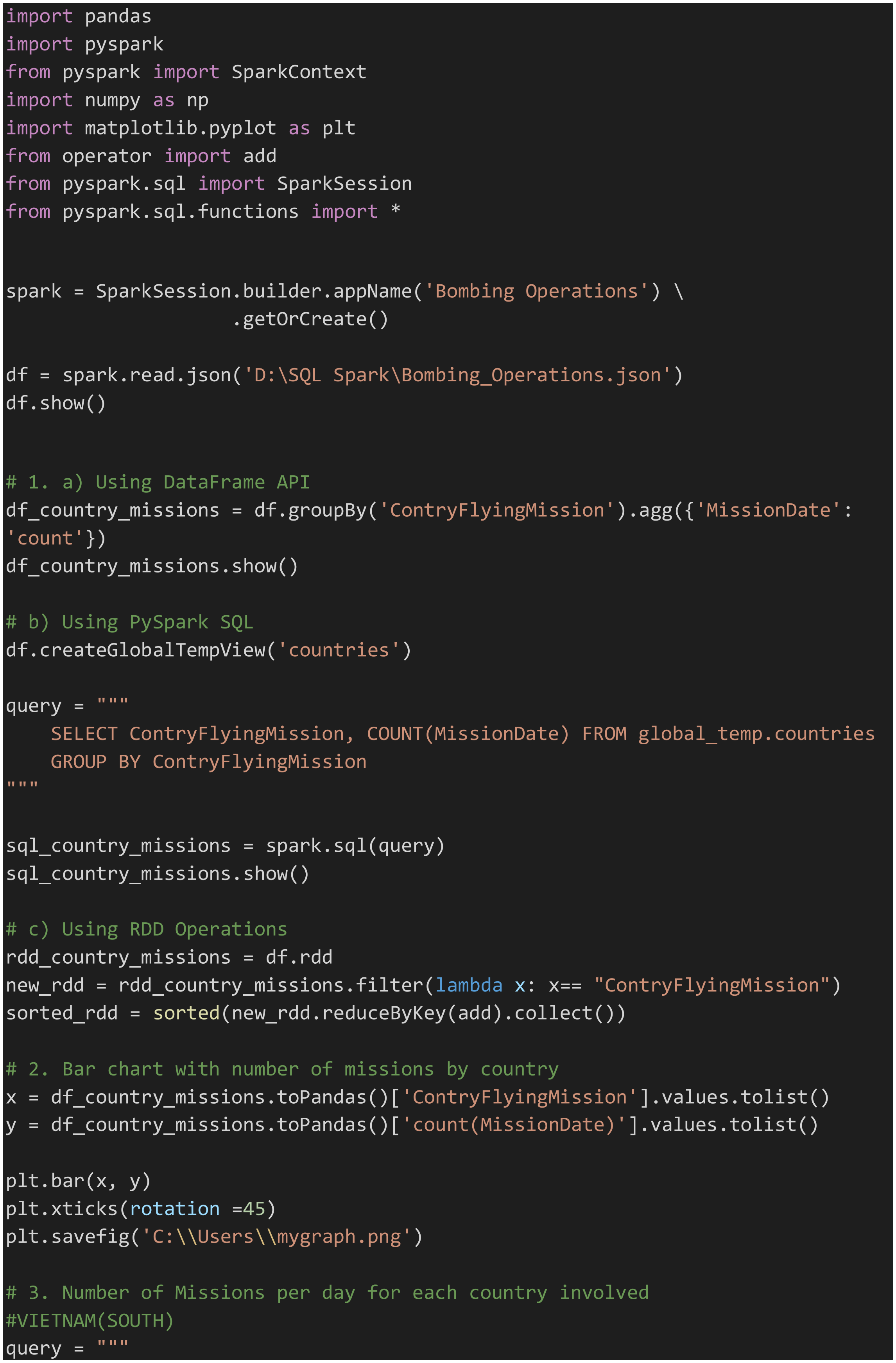
If the condition is not matched return the pair where the tuple x, has NULL

3. Grouping γ A,θ(B) (R). Given a relation R(A, B, C), with one grouping attribute A, one aggregated attribute B, and another attribute C, which is neither grouped or aggregated:
(a) Partition the tuples of R according to their values in attribute A.

Map prepares the grouping and the reduce completes the aggregation.

Map: For every tuple (a,b,c) return a key-value pair containing a and b

Reduce: With each key in place of a group
The key contains values of itself occurring throughout the tuples
Call every element of the list to the condition they are supposed to meet

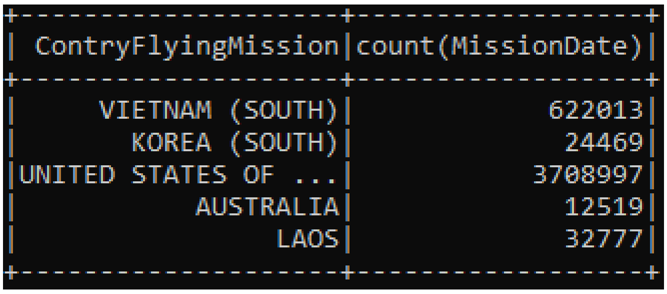


```
import pandas
import pyspark
from pyspark import SparkContext
import numpy as np
import matplotlib.pyplot as plt
from operator import add
from pyspark.sql import SparkSession
from pyspark.sql.functions import *


spark = SparkSession.builder.appName('Bombing Operations') \
                    .getOrCreate()

df = spark.read.json('D:\SQL Spark\Bombing_Operations.json')
df.show()


# 1. a) Using DataFrame API
df_country_missions = df.groupBy('ContryFlyingMission').agg({'MissionDate':
'count'})
df_country_missions.show()

# b) Using PySpark SQL
df.createGlobalTempView('countries')

query = """
    SELECT ContryFlyingMission, COUNT(MissionDate) FROM global_temp.countries
    GROUP BY ContryFlyingMission
"""

sql_country_missions = spark.sql(query)
sql_country_missions.show()

# c) Using RDD Operations
rdd_country_missions = df.rdd
new_rdd = rdd_country_missions.filter(lambda x: x== "ContryFlyingMission")
sorted_rdd = sorted(new_rdd.reduceByKey(add).collect())

# 2. Bar chart with number of missions by country
x = df_country_missions.toPandas()['ContryFlyingMission'].values.tolist()
y = df_country_missions.toPandas()['count(MissionDate)'].values.tolist()

plt.bar(x, y)
plt.xticks(rotation =45)
plt.savefig('C:\\Users\\mygraph.png')

# 3. Number of Missions per day for each country involved
#VIETNAM(SOUTH)
query = """
```

```
    SELECT ContryFlyingMission, MissionDate FROM global_temp.countries
    WHERE ContryFlyingMission = 'VIETNAM (SOUTH)'
"""

sql_country_missions = spark.sql(query)
sql_country_missions =
sql_country_missions.groupBy('MissionDate').agg({'MissionDate' : 'count'})


x = sql_country_missions.toPandas()['MissionDate'].values.tolist()
y = sql_country_missions.toPandas()['count(MissionDate)'].values.tolist()
plt.scatter(x, y)
plt.xlabel("Mission Dates")
plt.ylabel("Number of Missions")
plt.savefig('C:\\Users\\vietnam.png')

#KOREA (SOUTH)
query = """
    SELECT ContryFlyingMission, MissionDate FROM global_temp.countries
    WHERE ContryFlyingMission = 'KOREA (SOUTH)'
"""

sql_country_missions = spark.sql(query)
sql_country_missions =
sql_country_missions.groupBy('MissionDate').agg({'MissionDate' : 'count'})


x = sql_country_missions.toPandas()['MissionDate'].values.tolist()
y = sql_country_missions.toPandas()['count(MissionDate)'].values.tolist()
plt.scatter(x, y)
plt.xlabel("Mission Dates")
plt.ylabel("Number of Missions")
plt.savefig('C:\\Users\\korea.png')


#UNITED STATES OF AMERICA
query = """
    SELECT ContryFlyingMission, MissionDate FROM global_temp.countries
    WHERE ContryFlyingMission = 'UNITED STATES OF AMERICA'
"""

sql_country_missions = spark.sql(query)
sql_country_missions =
sql_country_missions.groupBy('MissionDate').agg({'MissionDate' : 'count'})


x = sql_country_missions.toPandas()['MissionDate'].values.tolist()
y = sql_country_missions.toPandas()['count(MissionDate)'].values.tolist()
plt.scatter(x, y)
plt.xlabel("Mission Dates")
plt.ylabel("Number of Missions")
plt.savefig('C:\\Users\\usa.png')


#LAOS
query = """
    SELECT ContryFlyingMission, MissionDate FROM global_temp.countries
    WHERE ContryFlyingMission = 'AUSTRALIA'
"""

sql_country_missions = spark.sql(query)
sql_country_missions =
sql_country_missions.groupBy('MissionDate').agg({'MissionDate' : 'count'})


x = sql_country_missions.toPandas()['MissionDate'].values.tolist()
y = sql_country_missions.toPandas()['count(MissionDate)'].values.tolist()
plt.scatter(x, y)
plt.xlabel("Mission Dates")
plt.ylabel("Number of Missions")
plt.savefig('C:\\User\\laos.png')

#AUSTRALIA
query = """
    SELECT ContryFlyingMission, MissionDate FROM global_temp.countries
    WHERE ContryFlyingMission = 'AUSTRALIA'
"""

sql_country_missions = spark.sql(query)
sql_country_missions =
sql_country_missions.groupBy('MissionDate').agg({'MissionDate' : 'count'})


x = sql_country_missions.toPandas()['MissionDate'].values.tolist()
y = sql_country_missions.toPandas()['count(MissionDate)'].values.tolist()
plt.scatter(x, y)
plt.xlabel("Mission Dates")
plt.ylabel("Number of Missions")
plt.savefig('C:\\Users\\australia.png')


# 4. Takeoffs launched to attack North Vietnam on 1966-06-29 from each
location
query_takeoffs = """
            SELECT TakeOffLocation, MissionDate, TargetCountry FROM
global_temp.countries
            WHERE MissionDate = '1966-06-29'
            AND TargetCountry = 'NORTH VIETNAM'
```

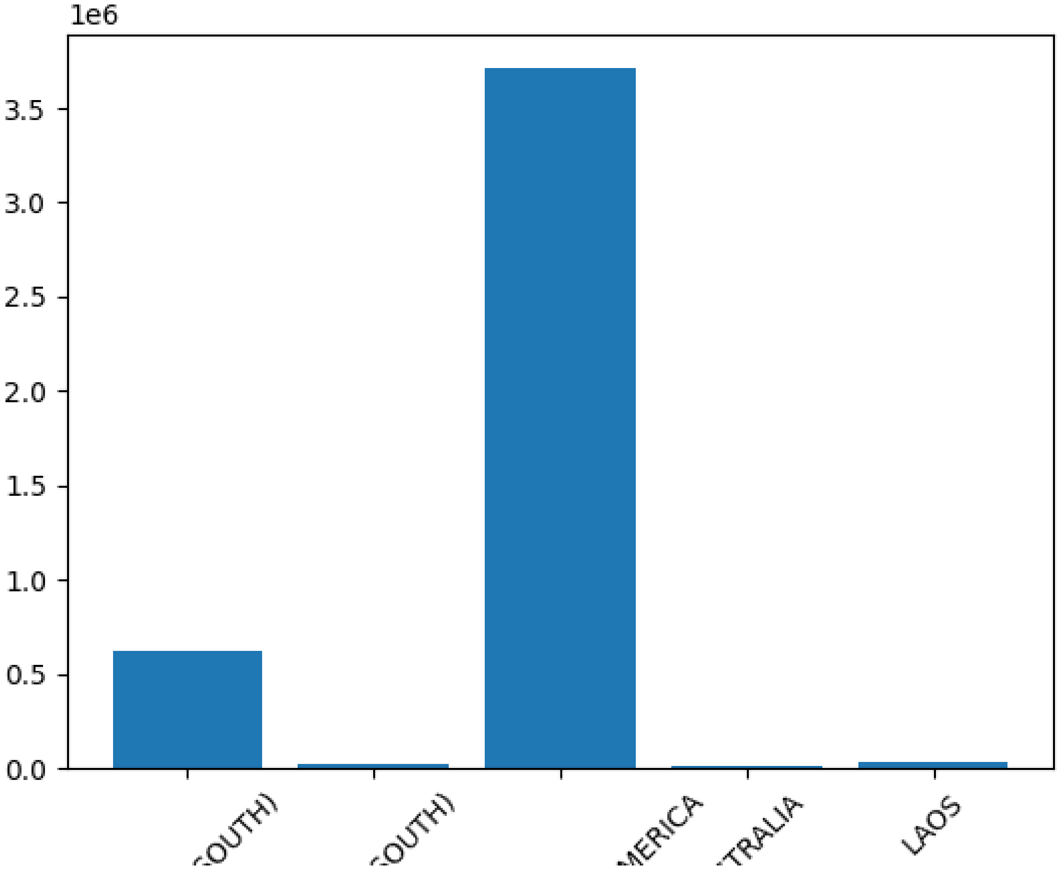
```
"""

takeoffs = spark.sql(query_takeoffs)
takeoffs = takeoffs.groupBy('TakeOffLocation').agg({'TargetCountry': 'count'})
takeoffs.show()

#5. Which Month saw the Highest Number Of Missions
mission_month = df.withColumn('MissionMonth', df.MissionDate[0:7])
mission_month.createOrReplaceTempView('missions')

query_missions = """
            SELECT MissionMonth, COUNT(MissionMonth) FROM missions
            GROUP BY MissionMonth
            ORDER BY count(MissionMonth) DESC
"""

missions = spark.sql(query_missions)
missions.show()

#6. Which campaigns saw the heaviest bombings
bombing_query = """
            SELECT ContryFlyingMission, MissionDate, TargetCountry,
WeaponsLoadedWeight FROM global_temp.countries
            ORDER BY WeaponsLoadedWeight DESC
"""

operation = spark.sql(bombing_query)
operation.show()


#7 Most Used Aircraft Type
df_aircrafts = spark.read.json('D:\SQL Spark\Aircraft_Glossary.json')
joined_df = df.join(df_aircrafts, df.AirCraft == df_aircrafts.AirCraft)
joined_df = joined_df.groupBy('AirCraftType').agg({'MissionDate': 'count'})
joined_df= joined_df.select('AirCraftType',
joined_df['count(MissionDate)'].alias('NumberMissions'))
joined_df.sort(joined_df.NumberMissions.desc()).show()
```

1. Query for Missions against Countries



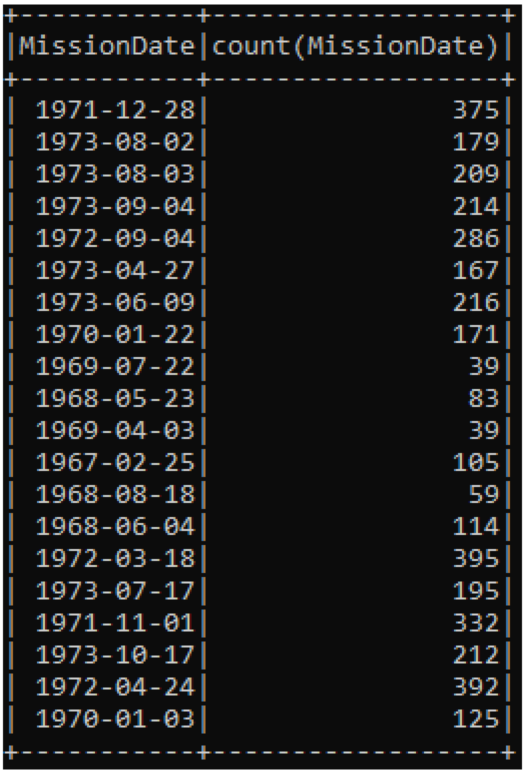
```
+-------------------+------------------+
| ContryFlyingMission|count(MissionDate)|
+-------------------+------------------+
|     VIETNAM (SOUTH)|            622013|
|       KOREA (SOUTH)|             24469|
|UNITED STATES OF ...|           3708997|
|           AUSTRALIA|             12519|
|                LAOS|             32777|
+-------------------+------------------+
```

2. Bar Chart



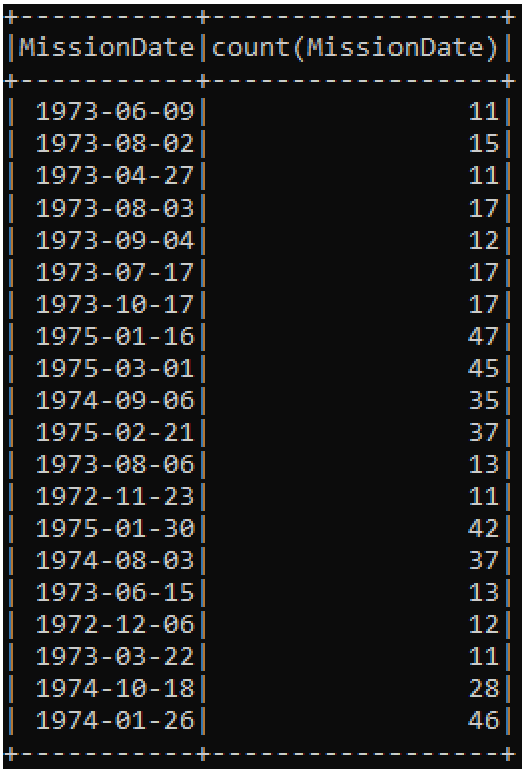
3. Number of missions for each country

A. SOUTH VIETNAM



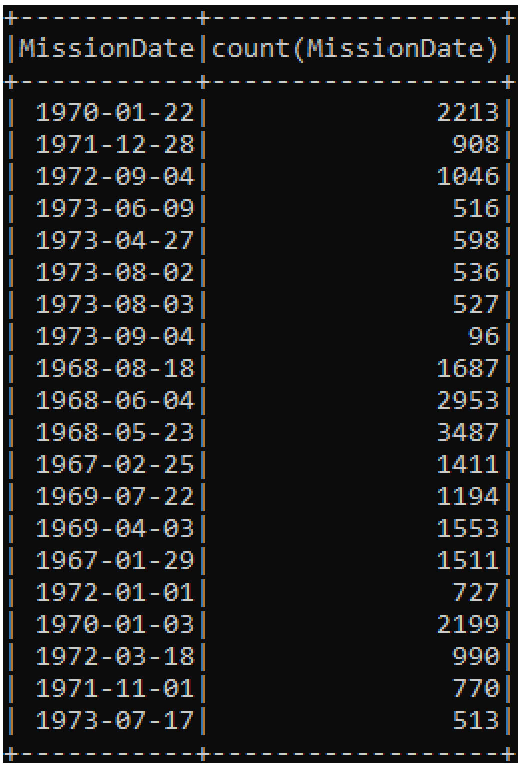
```
+-----------+------------------+
|MissionDate|count(MissionDate)|
+-----------+------------------+
| 1971-12-28|               375|
| 1973-08-02|               179|
| 1973-08-03|               209|
| 1973-09-04|               214|
| 1972-09-04|               286|
| 1973-04-27|               167|
| 1973-06-09|               216|
| 1970-01-22|               171|
| 1969-07-22|                39|
| 1968-05-23|                83|
| 1969-04-03|                39|
| 1967-02-25|               105|
| 1968-08-18|                59|
| 1968-06-04|               114|
| 1972-03-18|               395|
| 1973-07-17|               195|
| 1971-11-01|               332|
| 1973-10-17|               212|
| 1972-04-24|               392|
| 1970-01-03|               125|
+-----------+------------------+
```

B. SOUTH KOREA



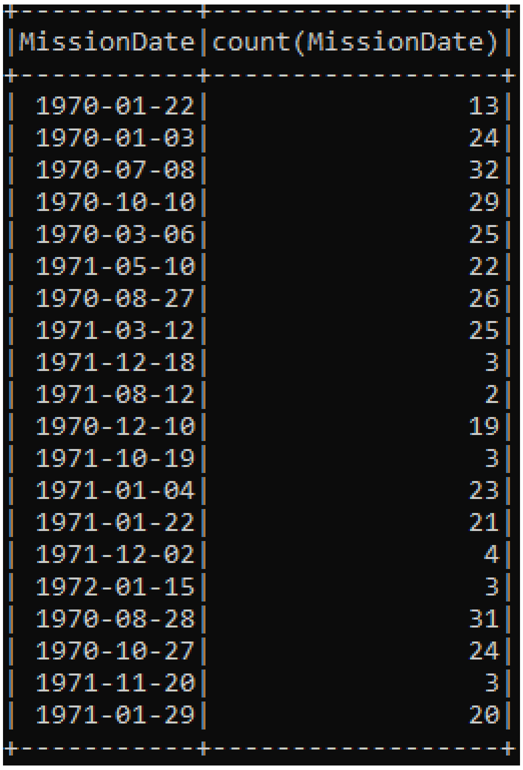
```
+-----------+------------------+
|MissionDate|count(MissionDate)|
+-----------+------------------+
| 1973-06-09|                11|
| 1973-08-02|                15|
| 1973-04-27|                11|
| 1973-08-03|                17|
| 1973-09-04|                12|
| 1973-07-17|                17|
| 1973-10-17|                17|
| 1975-01-16|                47|
| 1975-03-01|                45|
| 1974-09-06|                35|
| 1975-02-21|                37|
| 1973-08-06|                13|
| 1972-11-23|                11|
| 1975-01-30|                42|
| 1974-08-03|                37|
| 1973-06-15|                13|
| 1972-12-06|                12|
| 1973-03-22|                11|
| 1974-10-18|                28|
| 1974-01-26|                46|
+-----------+------------------+
```

C. UNITED STATES OF AMERICA



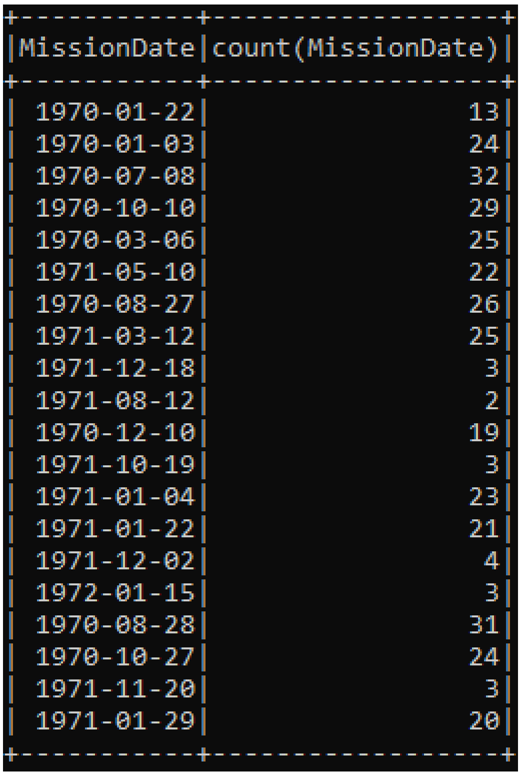
```
+-----------+------------------+
|MissionDate|count(MissionDate)|
+-----------+------------------+
| 1970-01-22|              2213|
| 1971-12-28|               908|
| 1972-09-04|              1046|
| 1973-06-09|               516|
| 1973-04-27|               598|
| 1973-08-02|               536|
| 1973-08-03|               527|
| 1973-09-04|                96|
| 1968-08-18|              1687|
| 1968-06-04|              2953|
| 1968-05-23|              3487|
| 1967-02-25|              1411|
| 1969-07-22|              1194|
| 1969-04-03|              1553|
| 1967-01-29|              1511|
| 1972-01-01|               727|
| 1970-01-03|              2199|
| 1972-03-18|               990|
| 1971-11-01|               770|
| 1973-07-17|               513|
+-----------+------------------+
```

D. AUSTRALIA



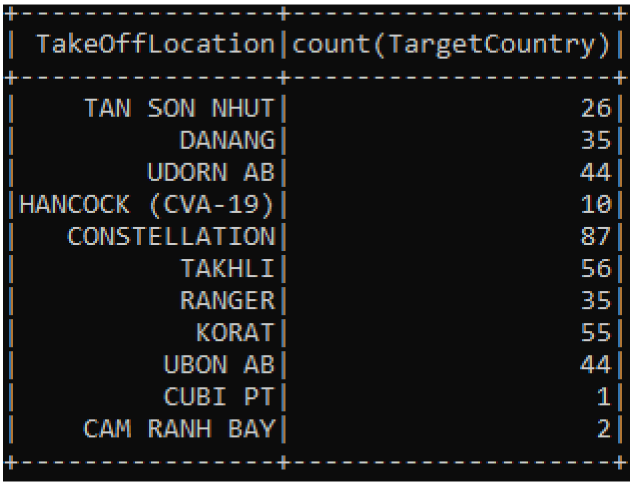
```
+-----------+------------------+
|MissionDate|count(MissionDate)|
+-----------+------------------+
| 1970-01-22|                13|
| 1970-01-03|                24|
| 1970-07-08|                32|
| 1970-10-10|                29|
| 1970-03-06|                25|
| 1971-05-10|                22|
| 1970-08-27|                26|
| 1971-03-12|                25|
| 1971-12-18|                 3|
| 1971-08-12|                 2|
| 1970-12-10|                19|
| 1971-10-19|                 3|
| 1971-01-04|                23|
| 1971-01-22|                21|
| 1971-12-02|                 4|
| 1972-01-15|                 3|
| 1970-08-28|                31|
| 1970-10-27|                24|
| 1971-11-20|                 3|
| 1971-01-29|                20|
+-----------+------------------+
```

E. LAOS



```
+-----------+------------------+
|MissionDate|count(MissionDate)|
+-----------+------------------+
| 1970-01-22|                13|
| 1970-01-03|                24|
| 1970-07-08|                32|
| 1970-10-10|                29|
| 1970-03-06|                25|
| 1971-05-10|                22|
| 1970-08-27|                26|
| 1971-03-12|                25|
| 1971-12-18|                 3|
| 1971-08-12|                 2|
| 1970-12-10|                19|
| 1971-10-19|                 3|
| 1971-01-04|                23|
| 1971-01-22|                21|
| 1971-12-02|                 4|
| 1972-01-15|                 3|
| 1970-08-28|                31|
| 1970-10-27|                24|
| 1971-11-20|                 3|
| 1971-01-29|                20|
+-----------+------------------+
```

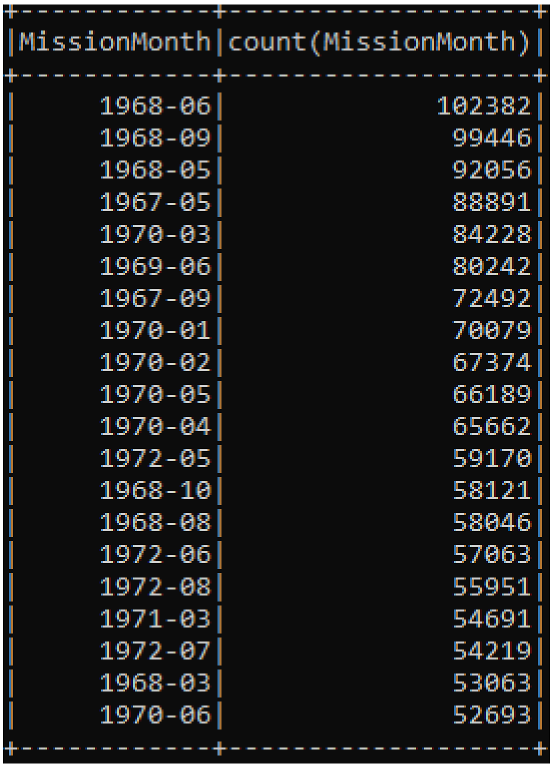
4. Takeoffs launched to attack North Vietnam on 1966-06-29 from each location



```
+----------------+--------------------+
| TakeOffLocation|count(TargetCountry)|
+----------------+--------------------+
|    TAN SON NHUT|                  26|
|          DANANG|                  35|
|        UDORN AB|                  44|
|HANCOCK (CVA-19)|                  10|
|   CONSTELLATION|                  87|
|          TAKHLI|                  56|
|          RANGER|                  35|
|           KORAT|                  55|
|         UBON AB|                  44|
|         CUBI PT|                   1|
|    CAM RANH BAY|                   2|
+----------------+--------------------+
```

5. Which month saw the highest number of missions

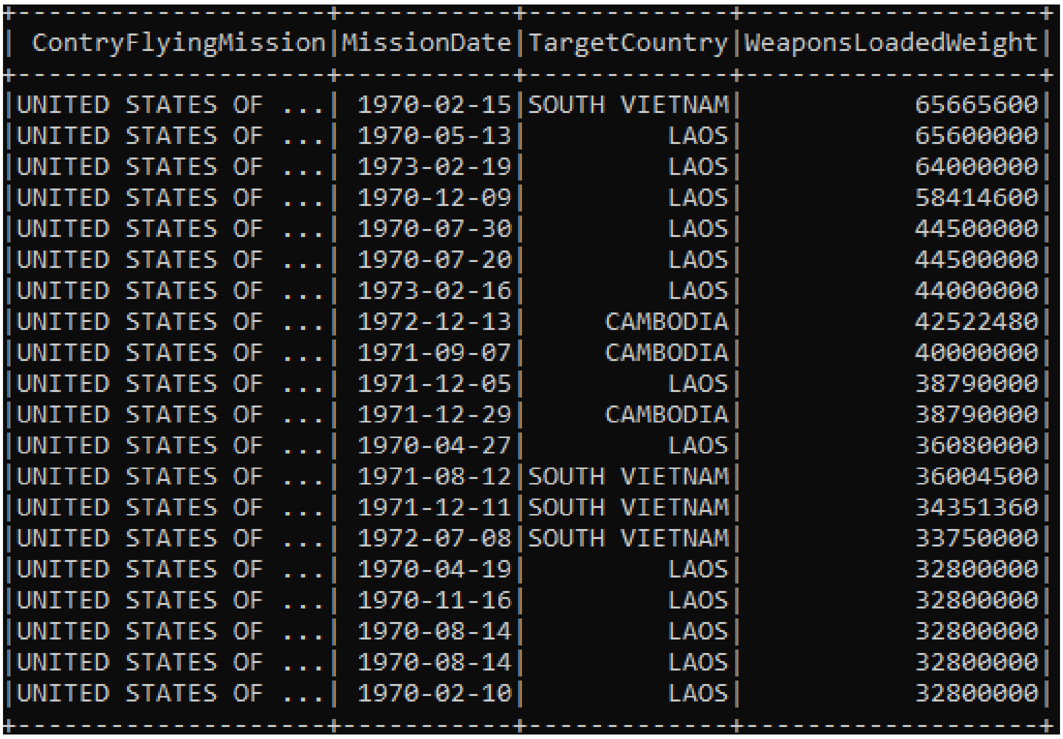
   June 1968 (102,382 Missions)



```
+------------+-------------------+
|MissionMonth|count(MissionMonth)|
+------------+-------------------+
|     1968-06|             102382|
|     1968-09|              99446|
|     1968-05|              92056|
|     1967-05|              88891|
|     1970-03|              84228|
|     1969-06|              80242|
|     1967-09|              72492|
|     1970-01|              70079|
|     1970-02|              67374|
|     1970-05|              66189|
|     1970-04|              65662|
|     1972-05|              59170|
|     1968-10|              58121|
|     1968-08|              58046|
|     1972-06|              57063|
|     1972-08|              55951|
|     1971-03|              54691|
|     1972-07|              54219|
|     1968-03|              53063|
|     1970-06|              52693|
+------------+-------------------+
```

6. Which campaigns saw the heaviest bombings

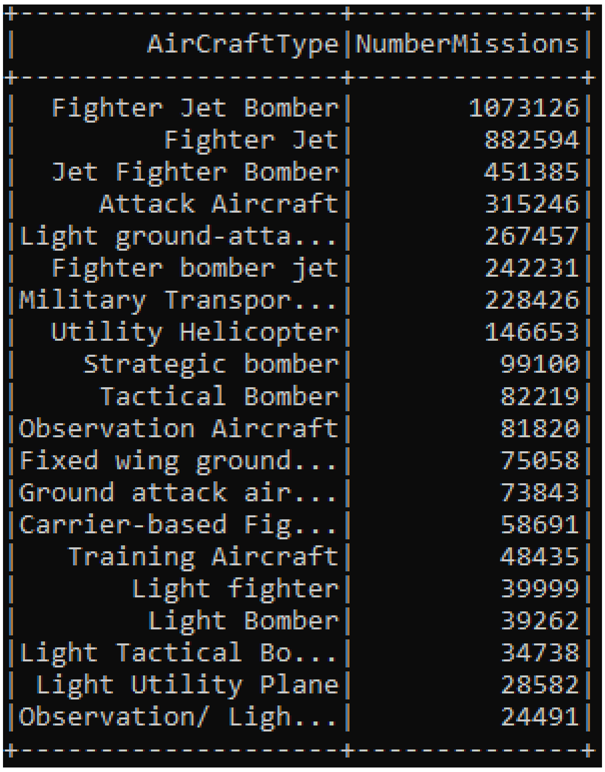
   The 1970-02-15 USA mission campaign targeting South Vietnam (Weight: 65665600)



```
+--------------------+-----------+-------------+-------------------+
| ContryFlyingMission|MissionDate|TargetCountry|WeaponsLoadedWeight|
+--------------------+-----------+-------------+-------------------+
|UNITED STATES OF ...| 1970-02-15|SOUTH VIETNAM|           65665600|
|UNITED STATES OF ...| 1970-05-13|         LAOS|           65600000|
|UNITED STATES OF ...| 1973-02-19|         LAOS|           64000000|
|UNITED STATES OF ...| 1970-12-09|         LAOS|           58414600|
|UNITED STATES OF ...| 1970-07-30|         LAOS|           44500000|
|UNITED STATES OF ...| 1970-07-20|         LAOS|           44500000|
|UNITED STATES OF ...| 1973-02-16|         LAOS|           44000000|
|UNITED STATES OF ...| 1972-12-13|     CAMBODIA|           42522480|
|UNITED STATES OF ...| 1971-09-07|     CAMBODIA|           40000000|
|UNITED STATES OF ...| 1971-12-05|         LAOS|           38790000|
|UNITED STATES OF ...| 1971-12-29|     CAMBODIA|           38790000|
|UNITED STATES OF ...| 1970-04-27|         LAOS|           36080000|
|UNITED STATES OF ...| 1971-08-12|SOUTH VIETNAM|           36004500|
|UNITED STATES OF ...| 1971-12-11|SOUTH VIETNAM|           34351360|
|UNITED STATES OF ...| 1972-07-08|SOUTH VIETNAM|           33750000|
|UNITED STATES OF ...| 1970-04-19|         LAOS|           32800000|
|UNITED STATES OF ...| 1970-11-16|         LAOS|           32800000|
|UNITED STATES OF ...| 1970-08-14|         LAOS|           32800000|
|UNITED STATES OF ...| 1970-08-14|         LAOS|           32800000|
|UNITED STATES OF ...| 1970-02-10|         LAOS|           32800000|
+--------------------+-----------+-------------+-------------------+
```

7. Most used aircraft type

   The Fighter Jet Bomber (1073126 Missions)



```
+--------------------+--------------+
|        AirCraftType|NumberMissions|
+--------------------+--------------+
|   Fighter Jet Bomber|       1073126|
|         Fighter Jet|        882594|
|  Jet Fighter Bomber|        451385|
|     Attack Aircraft|        315246|
|Light ground-atta...|        267457|
|   Fighter bomber jet|        242231|
|Military Transpor...|        228426|
|   Utility Helicopter|        146653|
|     Strategic bomber|         99100|
|      Tactical Bomber|         82219|
|Observation Aircraft|         81820|
|Fixed wing ground...|         75058|
|Ground attack air...|         73843|
|Carrier-based Fig...|         58691|
|    Training Aircraft|         48435|
|        Light fighter|         39999|
|         Light Bomber|         39262|
|Light Tactical Bo...|         34738|
|  Light Utility Plane|         28582|
|Observation/ Ligh...|         24491|
+--------------------+--------------+
```