Project Statement for Milestone 3

Group 6

Group member: Jinyang Ruan, Rusu Wu, Yi Yao, Brian Chan, Junqiao Mou

Project 2: Airline Search Engine (10%)

Airport and airline search (6%): **You must use PySpark for the following functions, no Pandas or other Python libraries are allowed.**

- (1%) Find list of airports operating in the Country X
- (1%) Find the list of Airlines having X stops
- (1%) List of airlines operating with code share
- (1%) Find the list of active airlines in the United States - Airline aggregation
- (1%) Find the country (or) territory has the highest number of Airports
- (1%) Top K cities with most incoming airlines

Answer:

Find list of airports operating in the Country X

```
airports_dat = spark.read.option("delimiter", ",").option("header", "false").csv("airports.dat").toDF("AirportID","Na
airports_dat.show(10)

airports = spark.sql("select * from airports where Country = 'United States'")
airports.show(10)
airports.createOrReplaceTempView("airports")
```

```
+---------+--------------------+------------+---------------+----+----+-----------------+-----------------+------
--+--------+---+--------------------+-------+-----------+
|AirportID|                Name|        City|        Country|IATA|ICAO|         Latitude|        Longitude|Altitu
de|Timezone|DST|                  Tz|   Type|     Source|
+---------+--------------------+------------+---------------+----+----+-----------------+-----------------+------
--+--------+---+--------------------+-------+-----------+
|        1|      Goroka Airport|      Goroka|Papua New Guinea| GKA|AYGA|-6.081689834590001|     145.391998291|    52
82|      10|  U|Pacific/Port_Moresby|airport|OurAirports|
|        2|      Madang Airport|      Madang|Papua New Guinea| MAG|AYMD|    -5.20707988739|     145.789001465|
20|      10|  U|Pacific/Port_Moresby|airport|OurAirports|
|        3|Mount Hagen Kagam...| Mount Hagen|Papua New Guinea| HGU|AYMH|-5.826789855957031|144.29600524902344|    53
88|      10|  U|Pacific/Port_Moresby|airport|OurAirports|
|        4|      Nadzab Airport|      Nadzab|Papua New Guinea| LAE|AYNZ|         -6.569803|        146.725977|     2
39|      10|  U|Pacific/Port_Moresby|airport|OurAirports|
|        5|Port Moresby Jack...|Port Moresby|Papua New Guinea| POM|AYPY|-9.443380355834961|147.22000122070312|     1
46|      10|  U|Pacific/Port_Moresby|airport|OurAirports|
|        6|Wewak Internation...|       Wewak|Papua New Guinea| WWK|AYWK|    -3.58383011818|     143.669006348|
19|      10|  U|Pacific/Port_Moresby|airport|OurAirports|
|        7|  Narsarsuaq Airport|Narssarssuaq|       Greenland| UAK|BGBW|     61.1604995728|    -45.4259986877|     1
12|      -3|  E|     America/Godthab|airport|OurAirports|
|        8|Godthaab / Nuuk A...|    Godthaab|       Greenland| GOH|BGGH|       64.19090271|    -51.6781005859|     2
83|      -3|  E|     America/Godthab|airport|OurAirports|
|        9|Kangerlussuaq Air...| Sondrestrom|       Greenland| SFJ|BGSF|     67.0122218992|    -50.7116031647|     1
65|      -3|  E|     America/Godthab|airport|OurAirports|
|       10|      Thule Air Base|       Thule|       Greenland| THU|BGTL|     76.5311965942|    -68.7032012939|     2
51|      -4|  E|       America/Thule|airport|OurAirports|
+---------+--------------------+------------+---------------+----+----+-----------------+-----------------+------
--+--------+---+--------------------+-------+-----------+
only showing top 10 rows
```

As the figure shows, list the airports operating in the United States.

```
+---------+--------------------+---------------+------------+----+----+-----------------+------------------+----
----+--------+---+----------------+-------+-----------+
|AirportID|                Name|           City|     Country|IATA|ICAO|         Latitude|         Longitude|Alti
tude|Timezone|DST|              Tz|   Type|     Source|
+---------+--------------------+---------------+------------+----+----+-----------------+------------------+----
----+--------+---+----------------+-------+-----------+
|     3411|Barter Island LRR...|  Barter Island|United States| BTI|PABA|     70.1340026855|     -143.582000732|
2|      -9|  A|America/Anchorage|airport|OurAirports|
|     3412|Wainwright Air St...| Fort Wainwright|United States|  \N|PAWT|       70.61340332|       -159.8600006|
35|      -9|  A|America/Anchorage|airport|OurAirports|
|     3413|Cape Lisburne LRR...|  Cape Lisburne|United States| LUR|PALU|       68.87509918|       -166.1100006|
16|      -9|  A|America/Anchorage|airport|OurAirports|
|     3414|Point Lay LRRS Ai...|      Point Lay|United States| PIZ|PPIZ|       69.73290253|       -163.0050049|
22|      -9|  A|America/Anchorage|airport|OurAirports|
|     3415|Hilo Internationa...|           Hilo|United States| ITO|PHTO|19.721399307250977|-155.04800415039062|
38|     -10|  N| Pacific/Honolulu|airport|OurAirports|
|     3416|Orlando Executive...|        Orlando|United States| ORL|KORL|           28.5455|         -81.332901|
113|      -5|  A| America/New_York|airport|OurAirports|
|     3417|     Bettles Airport|        Bettles|United States| BTT|PABT|       66.91390228|        -151.529007|
647|      -9|  A|America/Anchorage|airport|OurAirports|
|     3418|       Clear Airport|     Clear Mews|United States|  \N|PACL|         64.301201|        -149.119995|
552|      -9|  A|America/Anchorage|airport|OurAirports|
|     3419|Indian Mountain L...|Indian Mountains|United States| UTO|PAIM|       65.99279785|       -153.7039948|
1273|      -9|  A|America/Anchorage|airport|OurAirports|
|     3420|  Fort Yukon Airport|      Fort Yukon|United States| FYU|PFYU| 66.57150268554688|            -145.25|
433|      -9|  A|America/Anchorage|airport|OurAirports|
+---------+--------------------+---------------+------------+----+----+-----------------+------------------+----
----+--------+---+----------------+-------+-----------+
only showing top 10 rows
```

Find the list of Airlines having X stops

Firstly, list all the category values of attribute ‘Stops’.

```
routes_dat = spark.read.option("delimiter", ",").option("header", "false").csv("routes.dat").toDF("Airline","AirlineI
routes_dat.show(10)

routes = routes_dat.selectExpr("Airline","AirlineID","SourceAirport","SourceAirportID","DestinationAirport","Destinat
routes.groupBy("Stops").count().show(10)
routes.printSchema()
routes.createOrReplaceTempView("routes")
```

```
+-------+---------+-------------+---------------+-----------------+--------------------+---------+-----+---------+
|Airline|AirlineID|SourceAirport|SourceAirportID|DestinationAirport|DestinationAirportID|Codeshare|Stops|Equipment|
+-------+---------+-------------+---------------+-----------------+--------------------+---------+-----+---------+
|     2B|      410|          AER|           2965|              KZN|                2990|     null|    0|      CR2|
|     2B|      410|          ASF|           2966|              KZN|                2990|     null|    0|      CR2|
|     2B|      410|          ASF|           2966|              MRV|                2962|     null|    0|      CR2|
|     2B|      410|          CEK|           2968|              KZN|                2990|     null|    0|      CR2|
|     2B|      410|          CEK|           2968|              OVB|                4078|     null|    0|      CR2|
|     2B|      410|          DME|           4029|              KZN|                2990|     null|    0|      CR2|
|     2B|      410|          DME|           4029|              NBC|                6969|     null|    0|      CR2|
|     2B|      410|          DME|           4029|              TGK|                  \N|     null|    0|      CR2|
|     2B|      410|          DME|           4029|              UUA|                6160|     null|    0|      CR2|
|     2B|      410|          EGO|           6156|              KGD|                2952|     null|    0|      CR2|
+-------+---------+-------------+---------------+-----------------+--------------------+---------+-----+---------+
only showing top 10 rows

+-----+-----+
|Stops|count|
+-----+-----+
|    0|67652|
|    1|   11|
+-----+-----+
```

There are two categories.

Then, the list of Airlines having 1 stop will be.

```
routes_dat = spark.read.option("delimiter", ",").option("header", "false").csv("routes.dat").toDF("Airline","AirlineI
routes_dat.show(10)

routes = spark.sql("select * from routes where Stops = 1")
routes.show(10)
routes.printSchema()
routes.createOrReplaceTempView("routes")
```

```
+-------+---------+-------------+--------------+-----------------+--------------------+---------+-----+---------+
|Airline|AirlineID|SourceAirport|SourceAirportID|DestinationAirport|DestinationAirportID|Codeshare|Stops|Equipment|
+-------+---------+-------------+--------------+-----------------+--------------------+---------+-----+---------+
|     2B|      410|          AER|          2965|              KZN|                2990|     null|    0|      CR2|
|     2B|      410|          ASF|          2966|              KZN|                2990|     null|    0|      CR2|
|     2B|      410|          ASF|          2966|              MRV|                2962|     null|    0|      CR2|
|     2B|      410|          CEK|          2968|              KZN|                2990|     null|    0|      CR2|
|     2B|      410|          CEK|          2968|              OVB|                4078|     null|    0|      CR2|
|     2B|      410|          DME|          4029|              KZN|                2990|     null|    0|      CR2|
|     2B|      410|          DME|          4029|              NBC|                6969|     null|    0|      CR2|
|     2B|      410|          DME|          4029|              TGK|                  \N|     null|    0|      CR2|
|     2B|      410|          DME|          4029|              UUA|                6160|     null|    0|      CR2|
|     2B|      410|          EGO|          6156|              KGD|                2952|     null|    0|      CR2|
+-------+---------+-------------+--------------+-----------------+--------------------+---------+-----+---------+
only showing top 10 rows

+-------+---------+-------------+--------------+-----------------+--------------------+---------+-----+---------+
|Airline|AirlineID|SourceAirport|SourceAirportID|DestinationAirport|DestinationAirportID|Codeshare|Stops|Equipment|
+-------+---------+-------------+--------------+-----------------+--------------------+---------+-----+---------+
|     5T|     1623|          YRT|           132|              YEK|                  50|     null|    1|      ATR|
|     AC|      330|          ABJ|           253|              BRU|                 302|     null|    1|      333|
|     AC|      330|          YVR|           156|              YBL|                  30|     null|    1|      BEH|
|     CU|     1936|          FCO|          1555|              HAV|                1909|     null|    1|      767|
|     FL|     1316|          HOU|          3566|              SAT|                3621|     null|    1|      735|
|     FL|     1316|          MCO|          3878|              HOU|                3566|     null|    1|      73W|
|     FL|     1316|          MCO|          3878|              ORF|                3611|     null|    1|      717|
|     SK|     4319|          ARN|           737|              GEV|                 715|     null|    1|      ATP|
|     WN|     4547|          BOS|          3448|              MCO|                3878|     null|    1|      73W|
|     WN|     4547|          MCO|          3878|              BOS|                3448|     null|    1|      73W|
+-------+---------+-------------+--------------+-----------------+--------------------+---------+-----+---------+
only showing top 10 rows
```

List of airlines operating with code share

Firstly, list all the category values of attribute ‘Codeshare’.

```
routes_dat = spark.read.option("delimiter", ",").option("header", "false").csv("routes.dat").toDF("Airline","AirlineI
routes_dat.show(10)

routes = routes_dat.selectExpr("Airline","AirlineID","SourceAirport","SourceAirportID","DestinationAirport","Destinat
routes.groupBy("Codeshare").count().show(10)
routes.printSchema()
routes.createOrReplaceTempView("routes")
```

```
+-------+---------+-------------+--------------+-----------------+--------------------+---------+-----+---------+
|Airline|AirlineID|SourceAirport|SourceAirportID|DestinationAirport|DestinationAirportID|Codeshare|Stops|Equipment|
+-------+---------+-------------+--------------+-----------------+--------------------+---------+-----+---------+
|     2B|      410|          AER|          2965|              KZN|                2990|     null|    0|      CR2|
|     2B|      410|          ASF|          2966|              KZN|                2990|     null|    0|      CR2|
|     2B|      410|          ASF|          2966|              MRV|                2962|     null|    0|      CR2|
|     2B|      410|          CEK|          2968|              KZN|                2990|     null|    0|      CR2|
|     2B|      410|          CEK|          2968|              OVB|                4078|     null|    0|      CR2|
|     2B|      410|          DME|          4029|              KZN|                2990|     null|    0|      CR2|
|     2B|      410|          DME|          4029|              NBC|                6969|     null|    0|      CR2|
|     2B|      410|          DME|          4029|              TGK|                  \N|     null|    0|      CR2|
|     2B|      410|          DME|          4029|              UUA|                6160|     null|    0|      CR2|
|     2B|      410|          EGO|          6156|              KGD|                2952|     null|    0|      CR2|
+-------+---------+-------------+--------------+-----------------+--------------------+---------+-----+---------+
only showing top 10 rows
```

```
+---------+-----+
|Codeshare|count|
+---------+-----+
|     null|53066|
|        Y|14597|
+---------+-----+
```

There are two categories.

Then, the list of Airlines operating with code share will be.

```
routes_dat = spark.read.option("delimiter", ",").option("header", "false").csv("routes.dat").toDF("Airline","AirlineI
routes_dat.show(10)

routes = routes_dat.selectExpr("Airline","AirlineID","SourceAirport","SourceAirportID","DestinationAirport","Destinat
routes = spark.sql("select * from routes where Codeshare = 'Y'")
routes.show(10)
routes.createOrReplaceTempView("routes")
```

```
+-------+---------+-------------+---------------+------------------+--------------------+---------+-----+---------+
|Airline|AirlineID|SourceAirport|SourceAirportID|DestinationAirport|DestinationAirportID|Codeshare|Stops|Equipment|
+-------+---------+-------------+---------------+------------------+--------------------+---------+-----+---------+
|     2B|      410|          AER|           2965|               KZN|                2990|     null|    0|      CR2|
|     2B|      410|          ASF|           2966|               KZN|                2990|     null|    0|      CR2|
|     2B|      410|          ASF|           2966|               MRV|                2962|     null|    0|      CR2|
|     2B|      410|          CEK|           2968|               KZN|                2990|     null|    0|      CR2|
|     2B|      410|          CEK|           2968|               OVB|                4078|     null|    0|      CR2|
|     2B|      410|          DME|           4029|               KZN|                2990|     null|    0|      CR2|
|     2B|      410|          DME|           4029|               NBC|                6969|     null|    0|      CR2|
|     2B|      410|          DME|           4029|               TGK|                  \N|     null|    0|      CR2|
|     2B|      410|          DME|           4029|               UUA|                6160|     null|    0|      CR2|
|     2B|      410|          EGO|           6156|               KGD|                2952|     null|    0|      CR2|
+-------+---------+-------------+---------------+------------------+--------------------+---------+-----+---------+
only showing top 10 rows

+-------+---------+-------------+---------------+------------------+--------------------+---------+-----+---------+
|Airline|AirlineID|SourceAirport|SourceAirportID|DestinationAirport|DestinationAirportID|Codeshare|Stops|Equipment|
+-------+---------+-------------+---------------+------------------+--------------------+---------+-----+---------+
|     2P|      897|          GES|           2402|               MNL|                2397|        Y|    0|      320|
|     2P|      897|          MNL|           2397|               GES|                2402|        Y|    0|      320|
|     4M|     3201|          DFW|           3670|               EZE|                3988|        Y|    0|      777|
|     4M|     3201|          EZE|           3988|               DFW|                3670|        Y|    0|      777|
|     4M|     3201|          EZE|           3988|               JFK|                3797|        Y|    0|      777|
|     4M|     3201|          JFK|           3797|               EZE|                3988|        Y|    0|      777|
|     5N|      503|          ARH|           4362|               CSH|                6110|        Y|    0|      AN4|
|     5N|      503|          ARH|           4362|               MMK|                2949|        Y|    0|      AN4|
|     5N|      503|          ARH|           4362|               USK|                4369|        Y|    0|      AN4|
|     5N|      503|          CSH|           6110|               ARH|                4362|        Y|    0|      AN4|
+-------+---------+-------------+---------------+------------------+--------------------+---------+-----+---------+
only showing top 10 rows
```

Find the list of active airlines in the United States - Airline aggregation

List the table of airlines where the airlines are in the United States and active.

```
airlines_dat = spark.read.option("delimiter", ",").option("header", "false").csv("airlines.dat").toDF("AirlineID","Na
airlines_dat.show(10)

airlines = airlines_dat.selectExpr("AirlineID","Name","Alias","ITAT","ICAO","Callsign","Country","Active")
airlines = spark.sql("select * from airlines where Country = 'United States' and Active = 'Y'")
airlines.show(10)
airlines.createOrReplaceTempView("airlines")
```

```
+---------+--------------------+-----+----+----+-------------+-------------+------+
|AirlineID|                Name|Alias|ITAT|ICAO|     Callsign|      Country|Active|
+---------+--------------------+-----+----+----+-------------+-------------+------+
|       -1|             Unknown|   \N|   -| N/A|           \N|           \N|     Y|
|        1|      Private flight|   \N|   -| N/A|         null|         null|     Y|
|        2|         135 Airways|   \N|null| GNL|      GENERAL|United States|     N|
|        3|       1Time Airline|   \N|  1T| RNX|     NEXTIME| South Africa|     Y|
|        4|2 Sqn No 1 Elemen...|   \N|null| WYT|         null|United Kingdom|     N|
|        5|     213 Flight Unit|   \N|null| TFU|         null|       Russia|     N|
|        6|223 Flight Unit S...|   \N|null| CHD|CHKALOVSK-AVIA|       Russia|     N|
|        7|   224th Flight Unit|   \N|null| TTF|    CARGO UNIT|       Russia|     N|
|        8|         247 Jet Ltd|   \N|null| TWF| CLOUD RUNNER|United Kingdom|     N|
|        9|         3D Aviation|   \N|null| SEC|      SECUREX|United States|     N|
+---------+--------------------+-----+----+----+-------------+-------------+------+
only showing top 10 rows

+---------+--------------------+-----+----+----+-------------+-------------+------+
|AirlineID|                Name|Alias|ITAT|ICAO|     Callsign|      Country|Active|
+---------+--------------------+-----+----+----+-------------+-------------+------+
|       10|         40-Mile Air|   \N|  Q5| MLA|    MILE-AIR|United States|     Y|
|       22|      Aloha Airlines|   \N|  AQ| AAH|        ALOHA|United States|     Y|
|       24|   American Airlines|   \N|  AA| AAL|     AMERICAN|United States|     Y|
|       35|       Allegiant Air|   \N|  G4| AAY|    ALLEGIANT|United States|     Y|
|      109|Alaska Central Ex...|   \N|  KO| AER|      ACE AIR|United States|     Y|
|      149|  Air Cargo Carriers|   \N|  2Q| SNC|  NIGHT CARGO|United States|     Y|
|      210|Airlift Internati...|   \N|null| AIR|      AIRLIFT|United States|     Y|
|      281|America West Airl...|   \N|  HP| AWE|       CACTUS|United States|     Y|
|      282|       Air Wisconsin|   \N|  ZW| AWI|AIR WISCONSIN|United States|     Y|
|      287|Allegheny Commute...|   \N|null| ALO|    ALLEGHENY|United States|     Y|
+---------+--------------------+-----+----+----+-------------+-------------+------+
only showing top 10 rows
```

Find the country (or) territory has the highest number of Airports

Firstly, count the number of airports by country.

```
airports_dat = spark.read.option("delimiter", ",").option("header", "false").csv("airports.dat").toDF("AirportID","Na
airports_dat.show(10)


airports = airports_dat.selectExpr("AirportID","Name","City","Country","IATA","ICAO","Latitude","Longitude","Altitude
airports = spark.sql("select Country, COUNT(Country) as Number from airports GROUP BY Country")
airports.show(10)
airports.createOrReplaceTempView("airports")
```

```
--+--------+---+--------------------+-------+-----------+
|AirportID|                Name|        City|         Country|IATA|ICAO|          Latitude|          Longitude|Altitu
de|Timezone|DST|                  Tz|   Type|     Source|
+---------+--------------------+------------+----------------+----+----+------------------+-------------------+------
--+--------+---+--------------------+-------+-----------+
|        1|      Goroka Airport|      Goroka|Papua New Guinea| GKA|AYGA|-6.081689834590001|     145.391998291|    52
82|      10|  U|Pacific/Port_Moresby|airport|OurAirports|
|        2|      Madang Airport|      Madang|Papua New Guinea| MAG|AYMD|    -5.20707988739|     145.789001465|
20|      10|  U|Pacific/Port_Moresby|airport|OurAirports|
|        3|Mount Hagen Kagam...| Mount Hagen|Papua New Guinea| HGU|AYMH|-5.826789855957031|144.29600524902344|    53
88|      10|  U|Pacific/Port_Moresby|airport|OurAirports|
|        4|      Nadzab Airport|      Nadzab|Papua New Guinea| LAE|AYNZ|         -6.569803|        146.725977|     2
39|      10|  U|Pacific/Port_Moresby|airport|OurAirports|
|        5|Port Moresby Jack...|Port Moresby|Papua New Guinea| POM|AYPY|-9.443380355834961|147.22000122070312|     1
46|      10|  U|Pacific/Port_Moresby|airport|OurAirports|
|        6|Wewak Internation...|       Wewak|Papua New Guinea| WWK|AYWK|    -3.58383011818|     143.669006348|
19|      10|  U|Pacific/Port_Moresby|airport|OurAirports|
|        7|  Narsarsuaq Airport|Narssarssuaq|       Greenland| UAK|BGBW|     61.1604995728|    -45.4259986877|     1
12|      -3|  E|     America/Godthab|airport|OurAirports|
|        8|Godthaab / Nuuk A...|    Godthaab|       Greenland| GOH|BGGH|       64.19090271|    -51.6781005859|     2
83|      -3|  E|     America/Godthab|airport|OurAirports|
|        9|Kangerlussuaq Air...| Sondrestrom|       Greenland| SFJ|BGSF|     67.0122218992|    -50.7116031647|     1
65|      -3|  E|     America/Godthab|airport|OurAirports|
|       10|      Thule Air Base|       Thule|       Greenland| THU|BGTL|     76.5311965942|    -68.7032012939|     2
51|      -4|  E|       America/Thule|airport|OurAirports|
+---------+--------------------+------------+----------------+----+----+------------------+-------------------+------
--+--------+---+--------------------+-------+-----------+
only showing top 10 rows

+--------------------+------+
|             Country|Number|
+--------------------+------+
|                Chad|     6|
|            Paraguay|     9|
|            Anguilla|     1|
|              Russia|   264|
|British Indian Oc...|     1|
|               Yemen|    11|
|             Senegal|    11|
|              Sweden|    77|
|            Kiribati|    18|
|              Guyana|    13|
+--------------------+------+
only showing top 10 rows
```

Then, find the maximum number of the column of Number. And list the Name of the maximum number country (or) territory.

```
airports_Number = spark.sql("select MAX(airports.Number) from airports")
airports_Number.show()
```

```
+-----------+
|max(Number)|
+-----------+
|       1512|
+-----------+
```

```
airports_Number = spark.sql("select airports.Country, airports.Number from airports where airports.Number = 1512")
airports_Number.show()
```

```
+-------------+------+
|      Country|Number|
+-------------+------+
|United States|  1512|
+-------------+------+
```

Another method, order the result of count number then find the country (or) territory has the highest number of Airports which is United States.

```
airports_Number = spark.sql("select * from airports ORDER BY Number DESC")
airports_Number.show()
```

```
+--------------+------+
|       Country|Number|
+--------------+------+
| United States|  1512|
|        Canada|   430|
|     Australia|   334|
|        Brazil|   264|
|        Russia|   264|
|       Germany|   249|
|         China|   241|
|        France|   217|
|United Kingdom|   167|
|         India|   148|
|     Indonesia|   145|
|         Japan|   123|
|  South Africa|    99|
|     Argentina|    96|
|        Mexico|    84|
|         Italy|    83|
|          Iran|    82|
|        Sweden|    77|
|        Turkey|    76|
|      Colombia|    75|
+--------------+------+
only showing top 20 rows
```

Top K cities with most incoming airlines

Firstly, count the number of times each AirlineID appears in each DestinationAirportID. (Notice!!! The duplicate AirlineID of each DestinationAirportID category will be count into result.)

```
IncomeAirlinesList = spark.sql("select AirlineID, DestinationAirportID, COUNT(AirlineID) from routes GROUP BY Airline
IncomeAirlinesList.show(10)
```

```
+---------+--------------------+----------------+
|AirlineID|DestinationAirportID|count(AirlineID)|
+---------+--------------------+----------------+
|     4951|                1701|             219|
|     2009|                3682|             209|
|       24|                3670|             181|
|     5265|                3670|             177|
|     3320|                 340|             169|
|      137|                1382|             164|
|     5209|                3550|             161|
|      137|                3682|             161|
|     3090|                3682|             160|
|     5209|                3830|             158|
+---------+--------------------+----------------+
only showing top 10 rows
```

Beacuse each AirlineID will be appear once by each DestinationAirportID category. (Duplicate AirlineID will be statistic as number shown in the table) Then, we can get the final result by counting DestinationAirportID which is no duplicate AirlineID.

```
TopCities = spark.sql("select DestinationAirportID, COUNT(DestinationAirportID) as Number from (select AirlineID, Des
TopCities.show(10)
```

```
[Stage 1046:================================================>  (191 + 4) / 200]
```

```
+--------------------+------+
|DestinationAirportID|Number|
+--------------------+------+
|                1382|   109|
|                 340|   100|
|                3885|    98|
|                1555|    92|
|                 507|    86|
|                3077|    84|
|                2188|    84|
|                3316|    83|
|                 580|    80|
|                3364|    80|
+--------------------+------+
only showing top 10 rows
```

Verification

There are 12 AirlineID where its DestinatinAirportID is 2554. However, there are only four different incoming airlines. That satisfy the query result.

```
TopCities = spark.sql("select DestinationAirportID, COUNT(DestinationAirportID) as Number from (select AirlineID, Des
TopCities.show(1000)
```

```
|                5562|     4|
|                 498|     4|
|                3074|     4|
|                3319|     4|
|                4124|     4|
|                6133|     4|
|                6493|     4|
|                2792|     4|
|                9310|     4|
|                4214|     4|
|                4367|     4|
|                1918|     4|
|                3057|     4|
|                2554|     4|
+-------------------+------+
only showing top 1000 rows
```

```
airline_quary = spark.sql("select AirlineID, COUNT(AirlineID) from routes where DestinationAirportID = 2554 GROUP BY
airline_quary.show(60)
```

```
+---------+----------------+
|AirlineID|count(AirlineID)|
+---------+----------------+
|     3200|               1|
|     4867|               5|
|     1790|               3|
|    13983|               3|
+---------+----------------+
```

Finally, join the 'City' column from airports data with the result. Therefore, the Top 10 cities with most incoming airlines as figure shows below.

```
TopCities = spark.sql("select TopAirportID.DestinationAirportID, TopAirportID.Number, airports.AirportID, airports.Ci
TopCities.show(10)
```

```
[Stage 1076:================================================>  (192 + 4) / 200]
```

```
+-------------------+------+---------+---------+
|DestinationAirportID|Number|AirportID|     City|
+-------------------+------+---------+---------+
|               1382|   109|     1382|    Paris|
|                340|   100|      340|Frankfurt|
|               3885|    98|     3885|  Bangkok|
|               1555|    92|     1555|     Rome|
|                507|    86|      507|   London|
|               2188|    84|     2188|    Dubai|
|               3077|    84|     3077|Hong Kong|
|               3316|    83|     3316|Singapore|
|               3364|    80|     3364|  Beijing|
|                580|    80|      580|Amsterdam|
+-------------------+------+---------+---------+
only showing top 10 rows
```

Geospatial analytics (4%): **You must use Apache Sedona (https://sedona.apache.org/), no GeoPandas or other Python libraries are allowed**

- (2%) Find the closest airport to a city X's geospatial coordinate
- (2%) Find the airport in each US state's geospatial boundary

Answer:

Find the closest airport to a city X's geospatial coordinate

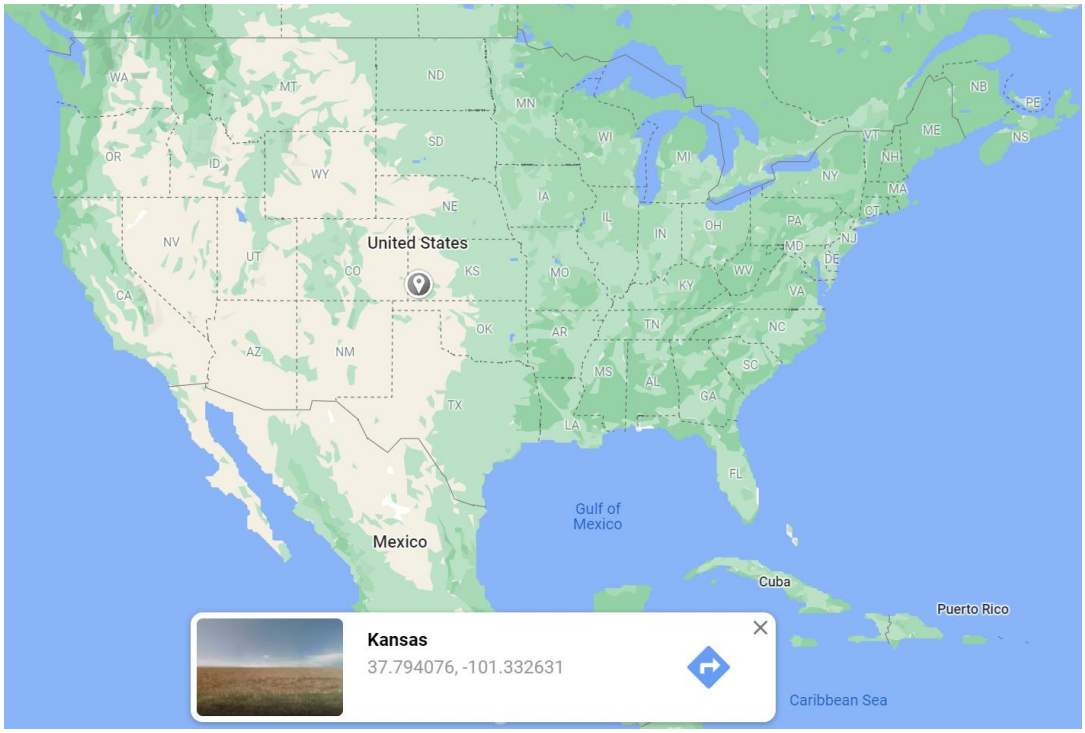
Firstly, combine the Latitude and Longitude column to a geometry column named 'Point'.

```
airports_point = spark.sql("select *, ST_Point(CAST(Latitude as Double), CAST(Longitude as Double)) as Point from air
airports_point.show(5)
airports_point.printSchema()
airports_point.createOrReplaceTempView("airports_point")
```

```
+---------+--------------------+------------+---------------+----+----+------------------+------------------+------
--+--------+---+--------------------+-------+----------+--------------------+
|AirportID|                Name|        City|        Country|IATA|ICAO|          Latitude|         Longitude|Altitu
de|Timezone|DST|                  Tz|   Type|    Source|               Point|
+---------+--------------------+------------+---------------+----+----+------------------+------------------+------
--+--------+---+--------------------+-------+----------+--------------------+
|        1|      Goroka Airport|      Goroka|Papua New Guinea| GKA|AYGA|-6.081689834590001|     145.391998291|    52
82|      10|  U|Pacific/Port_Moresby|airport|OurAirports|POINT (-6.0816898...|
|        2|      Madang Airport|      Madang|Papua New Guinea| MAG|AYMD|    -5.20707988739|     145.789001465|
20|      10|  U|Pacific/Port_Moresby|airport|OurAirports|POINT (-5.2070798...|
|        3|Mount Hagen Kagam...| Mount Hagen|Papua New Guinea| HGU|AYMH|-5.826789855957031|144.29600524902344|    53
88|      10|  U|Pacific/Port_Moresby|airport|OurAirports|POINT (-5.8267898...|
|        4|      Nadzab Airport|      Nadzab|Papua New Guinea| LAE|AYNZ|         -6.569803|        146.725977|     2
39|      10|  U|Pacific/Port_Moresby|airport|OurAirports|POINT (-6.569803 ...|
|        5|Port Moresby Jack...|Port Moresby|Papua New Guinea| POM|AYPY|-9.443380355834961|147.22000122070312|     1
46|      10|  U|Pacific/Port_Moresby|airport|OurAirports|POINT (-9.4433803...|
+---------+--------------------+------------+---------------+----+----+------------------+------------------+------
--+--------+---+--------------------+-------+----------+--------------------+
only showing top 5 rows

root
 |-- AirportID: string (nullable = true)
 |-- Name: string (nullable = true)
 |-- City: string (nullable = true)
 |-- Country: string (nullable = true)
 |-- IATA: string (nullable = true)
 |-- ICAO: string (nullable = true)
 |-- Latitude: string (nullable = true)
 |-- Longitude: string (nullable = true)
 |-- Altitude: string (nullable = true)
 |-- Timezone: string (nullable = true)
 |-- DST: string (nullable = true)
 |-- Tz: string (nullable = true)
 |-- Type: string (nullable = true)
 |-- Source: string (nullable = true)
 |-- Point: geometry (nullable = false)
```

Then, randomly chose a point from the Google Map and get the point coordinate is (37.794076, -101.332631).

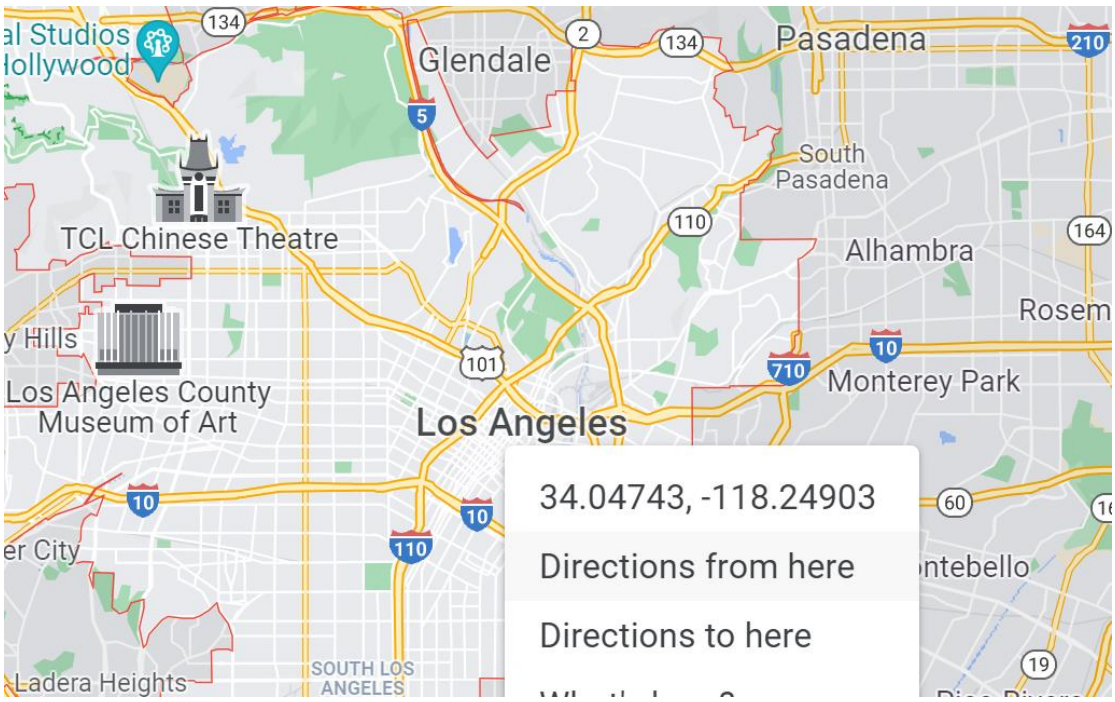


Finally, calculate the nearest the distance of airport close to the point is Ulysses Airport which Coordinate is (37.60400009, -101.3740005)

```
dist_to_Pullman = spark.sql("select Name, City, Latitude, Longitude, Point as Coordinate, ST_Distance(Point, ST_Point
dist_to_Pullman.show()
```

```
+--------------------+----------+-----------------+------------------+--------------------+-------------------+
|                Name|      City|         Latitude|         Longitude|          Coordinate|           Distance|
+--------------------+----------+-----------------+------------------+--------------------+-------------------+
|     Ulysses Airport|   Ulysses|       37.60400009|       -101.3740005|POINT (37.6040000...|0.19452580058330582|
|Sublette Municipa...|  Sublette|       37.49140167|       -100.8300018|POINT (37.4914016...| 0.5867263951226219|
|Garden City Regio...|Garden City|     37.9275016785|      -100.723999023|POINT (37.9275016...| 0.6230853032371386|
|Scott City Munici...| Scott City|38.474300384521484|-100.88500213623047|POINT (38.4743003...|  0.814295286107714|
|Liberal Mid-Ameri...|   Liberal|        37.0442009|       -100.9599991|POINT (37.0442009...| 0.8373572705229333|
|Elkhart Morton Co...|   Elkhart|         37.000702|        -101.879997|POINT (37.000702 ...| 0.9638733536269153|
|Meade Municipal A...|     Meade| 37.27690124511719|-100.35600280761719|POINT (37.2769012...|  1.105111918877462|
|Guymon Municipal ...|    Guymon|     36.6851005554|      -101.508003235|POINT (36.6851005...| 1.1227564105961092|
|Dodge City Region...| Dodge City| 37.76340103149414|  -99.9655990600586|POINT (37.7634010...|     1.3673760559966|
|Lamar Municipal A...|     Lamar|38.069698333699996|       -102.68800354|POINT (38.0696983...| 1.3831132972465632|
|Renner Field-Good...|  Goodland|       39.37060165|       -101.6989975|POINT (39.3706016...|  1.618535615125652|
|Hutchinson County...|    Borger|35.700901031499995|      -101.393997192|POINT (35.7009010...|  2.094074320141375|
|Dalhart Municipal...|   Dalhart|     36.0225982666|       -102.54699707|POINT (36.0225982...| 2.1477472644376085|
|La Junta Municipa...|  La Junta|       38.04970169|       -103.5090027|POINT (38.0497016...| 2.1913325329462947|
|  Perry Lefors Field|     Pampa|   35.612998962402|   -100.99600219727|POINT (35.6129989...| 2.2069018996694614|
|West Woodward Air...|  Woodward|           36.438|        -99.5226667|POINT (36.438 -99...| 2.2616173162253803|
|Flagler Aerial Sp...|   Flagler|39.279998779296875|-103.06700134277344|POINT (39.2799987...| 2.2838579185066195|
|Larned Pawnee Cou...|    Larned|       38.20859909|       -99.08599854|POINT (38.2085990...|  2.284553961382573|
|Hays Regional Air...|      Hays|       38.84220123|       -99.27320099|POINT (38.8422012...|  2.310804722137109|
|Mc Cook Ben Nelso...|   McCook|       40.20629883|       -100.5920029|POINT (40.2062988...|  2.523360648833784|
+--------------------+----------+-----------------+------------------+--------------------+-------------------+
only showing top 20 rows
```

One more example, chose Los Angeles City from the Google Earth and get the point coordinate is (34.04743, -118.24903).



Calculate the nearest the distance of airport close to the Los Angeles is Jack Northrop Field Airport which Coordinate is (33.922798, -118.334999).

```
dist_to_Pullman = spark.sql("select Name, City, Latitude, Longitude, Point as Coordinate, ST_Distance(Point, ST_Point
dist_to_Pullman.show(10)
```

```
+--------------------+------------+------------------+------------------+--------------------+--------------------+
|                Name|        City|          Latitude|         Longitude|          Coordinate|            Distance|
+--------------------+------------+------------------+------------------+--------------------+--------------------+
|Jack Northrop Fie...|   Hawthorne|         33.922798|       -118.334999|POINT (33.922798 ...|0.15140609097720645|
|    Bob Hope Airport|     Burbank| 34.20069885253906|-118.35900115966797|POINT (34.2006988...| 0.1886398608920134|
|Los Angeles Inter...| Los Angeles|       33.94250107|       -118.4079971|POINT (33.9425010...| 0.1904747207199797|
|Santa Monica Muni...|Santa Monica|34.015800476100004|     -118.450996399|POINT (34.0158004...|0.20442811232109792|
|San Gabriel Valle...|    El Monte|         34.086102|        -118.035004|POINT (34.086102 ...|0.21749172917608062|
|Long Beach /Daugh...|  Long Beach|       33.81769943|       -118.1520004|POINT (33.8176994...|0.24938098979009038|
|     Zamperini Field|    Torrance|   33.803398132324|   -118.33999633789|POINT (33.8033981...| 0.2604350726583375|
|    Whiteman Airport| Los Angeles|     34.2593002319|     -118.413002014|POINT (34.2593002...|0.26791009040455105|
|    Van Nuys Airport|    Van Nuys|   34.209800720215|   -118.48999786377|POINT (34.2098007...| 0.2905679991895415|
|Fullerton Municip...|   Fullerton|     33.8720016479|     -117.980003357|POINT (33.8720016...| 0.3211704241744135|
+--------------------+------------+------------------+------------------+--------------------+--------------------+
only showing top 10 rows
```

Find the airport in each US state's geospatial boundary

Firstly, load state boundries in WKT TSV and convert WKT string column to a geometry column.

```
states_wkt = spark.read.option("delimiter", "\t").option("header", "false").csv("boundary-each-state.tsv").toDF("s_na
states_wkt.show(5)
states_wkt.printSchema()

states = states_wkt.selectExpr("s_name", "ST_GeomFromWKT(s_bound) as s_bound")
states.show(5)
states.printSchema()
states.createOrReplaceTempView("states")
```

```
+----------+--------------------+
|    s_name|             s_bound|
+----------+--------------------+
|    Alaska|POLYGON((-141.020...|
|   Alabama|POLYGON((-88.1955...|
|  Arkansas|POLYGON((-94.0416...|
|   Arizona|POLYGON((-112.598...|
|California|POLYGON((-124.400...|
+----------+--------------------+
only showing top 5 rows

root
 |-- s_name: string (nullable = true)
 |-- s_bound: string (nullable = true)

+----------+--------------------+
|    s_name|             s_bound|
+----------+--------------------+
|    Alaska|POLYGON ((-141.02...|
|   Alabama|POLYGON ((-88.195...|
|  Arkansas|POLYGON ((-94.041...|
|   Arizona|POLYGON ((-112.59...|
|California|POLYGON ((-124.40...|
+----------+--------------------+
only showing top 5 rows

root
 |-- s_name: string (nullable = true)
 |-- s_bound: geometry (nullable = false)
```

Secondly, load city locations and convert the string column to a geometry column.

```
airports_point = spark.sql("select *, ST_Point(CAST(Longitude as Double), CAST(Latitude as Double)) as Point from air
airports_point.show(5)
airports_point.printSchema()
airports_point.createOrReplaceTempView("airports_point")
```

```
+---------+--------------------+------------+---------------+----+----+-----------------+-----------------+------
--+--------+---+--------------------+-------+-----------+--------------------+
|AirportID|                Name|        City|        Country|IATA|ICAO|         Latitude|        Longitude|Altitu
de|Timezone|DST|                  Tz|   Type|     Source|               Point|
+---------+--------------------+------------+---------------+----+----+-----------------+-----------------+------
--+--------+---+--------------------+-------+-----------+--------------------+
|        1|      Goroka Airport|      Goroka|Papua New Guinea| GKA|AYGA|-6.081689834590001|     145.391998291|    52
82|      10|  U|Pacific/Port_Moresby|airport|OurAirports|POINT (145.391998...|
|        2|      Madang Airport|      Madang|Papua New Guinea| MAG|AYMD|    -5.20707988739|     145.789001465|
20|      10|  U|Pacific/Port_Moresby|airport|OurAirports|POINT (145.789001...|
|        3|Mount Hagen Kagam...| Mount Hagen|Papua New Guinea| HGU|AYMH|-5.826789855957031|144.29600524902344|    53
88|      10|  U|Pacific/Port_Moresby|airport|OurAirports|POINT (144.296005...|
|        4|      Nadzab Airport|      Nadzab|Papua New Guinea| LAE|AYNZ|         -6.569803|        146.725977|     2
39|      10|  U|Pacific/Port_Moresby|airport|OurAirports|POINT (146.725977...|
|        5|Port Moresby Jack...|Port Moresby|Papua New Guinea| POM|AYPY|-9.443380355834961|147.22000122070312|     1
46|      10|  U|Pacific/Port_Moresby|airport|OurAirports|POINT (147.220001...|
+---------+--------------------+------------+---------------+----+----+-----------------+-----------------+------
--+--------+---+--------------------+-------+-----------+--------------------+
only showing top 5 rows

root
 |-- AirportID: string (nullable = true)
 |-- Name: string (nullable = true)
 |-- City: string (nullable = true)
 |-- Country: string (nullable = true)
 |-- IATA: string (nullable = true)
 |-- ICAO: string (nullable = true)
 |-- Latitude: string (nullable = true)
 |-- Longitude: string (nullable = true)
 |-- Altitude: string (nullable = true)
 |-- Timezone: string (nullable = true)
 |-- DST: string (nullable = true)
 |-- Tz: string (nullable = true)
 |-- Type: string (nullable = true)
 |-- Source: string (nullable = true)
 |-- Point: geometry (nullable = false)
```

Finally, we can get the list of the airport in each US state's geospatial boundary.

```
city_per_state = spark.sql("select states.s_name, states.s_bound, airports_point.Name, airports_point.City, airports_
city_per_state.show(100)
```

```
21/10/31 12:17:29 WARN JoinQuery: UseIndex is true, but no index exists. Will build index on the fly.
+-------+--------------------+--------------------+----------------+--------------------+
| s_name|             s_bound|                Name|            City|               Point|
+-------+--------------------+--------------------+----------------+--------------------+
|Alabama|POLYGON ((-88.195...|Mobile Downtown A...|          Mobile|POINT (-88.068099...|
|Alabama|POLYGON ((-88.195...|Thomas C Russell ...|  Alexander City|POINT (-85.962997...|
|Alabama|POLYGON ((-88.195...|Northwest Alabama...|   Muscle Shoals|POINT (-87.610198...|
|Alabama|POLYGON ((-88.195...|Cairns AAF (Fort ...|Fort Rucker/Ozark|POINT (-85.713401...|
|Alabama|POLYGON ((-88.195...|Evergreen Regiona...|       Evergreen|POINT (-87.043999...|
|Alabama|POLYGON ((-88.195...|Centre-Piedmont-C...|          Centre|POINT (-85.610069...|
|Alabama|POLYGON ((-88.195...|Pryor Field Regio...|         Decatur|POINT (-86.945396...|
|Alabama|POLYGON ((-88.195...|Northeast Alabama...|         Gadsden|POINT (-86.088996...|
|Alabama|POLYGON ((-88.195...|Anniston Regional...|        Anniston|POINT (-85.8581 3...|
|Alabama|POLYGON ((-88.195...|Mobile Regional A...|          Mobile|POINT (-88.242797...|
|Alabama|POLYGON ((-88.195...|Shelby County Air...|       Alabaster|POINT (-86.782798...|
|Alabama|POLYGON ((-88.195...|Maxwell Air Force...|      Montgomery|POINT (-86.365799...|
|Alabama|POLYGON ((-88.195...|Birmingham-Shuttl...|      Birmingham|POINT (-86.753501...|
|Alabama|POLYGON ((-88.195...|Jack Edwards Airport|     Gulf Shores|POINT (-87.671798...|
```

```
|Alabama|POLYGON ((-88.195...|Jack Edwards Airport|     Gulf Shores|POINT (-87.671798...|
|Alabama|POLYGON ((-88.195...|Redstone Army Air...|        Redstone|POINT (-86.684799...|
|Alabama|POLYGON ((-88.195...|Madison County Ex...|      Huntsville|POINT (-86.557502...|
|Alabama|POLYGON ((-88.195...|         Craig Field|           Selma|POINT (-86.987800...|
|Alabama|POLYGON ((-88.195...|        Weedon Field|          Eufala|POINT (-85.128898...|
|Alabama|POLYGON ((-88.195...|Auburn University...|          Auburn|POINT (-85.433998...|
|Alabama|POLYGON ((-88.195...|Lawson Army Air F...|    Fort Benning|POINT (-84.991302...|
|Alabama|POLYGON ((-88.195...|Huntsville Intern...|      Huntsville|POINT (-86.775100...|
|Alabama|POLYGON ((-88.195...|    Bessemer Airport|        Bessemer|POINT (-86.925903...|
|Alabama|POLYGON ((-88.195...|Talladega Municip...|       Talladega|POINT (-86.050903...|
|Alabama|POLYGON ((-88.195...|Montgomery Region...|      MONTGOMERY|POINT (-86.393997...|
|Alabama|POLYGON ((-88.195...|Troy Municipal Ai...|            Troy|POINT (-86.012101...|
|Alabama|POLYGON ((-88.195...|Merkel Field Syla...|       Sylacauga|POINT (-86.305496...|
|Alabama|POLYGON ((-88.195...|Dothan Regional A...|          Dothan|POINT (-85.449600...|
|Alabama|POLYGON ((-88.195...|Tuscaloosa Region...|   Tuscaloosa AL|POINT (-87.611396...|
|Alabama|POLYGON ((-88.195...|Enterprise Munici...|      Enterprise|POINT (-85.899902...|
| Alaska|POLYGON ((-141.02...|    Kivalina Airport|        Kivalina|POINT (-164.56300...|
| Alaska|POLYGON ((-141.02...|    Mekoryuk Airport|        Mekoryuk|POINT (-166.27099...|
| Alaska|POLYGON ((-141.02...|       Wales Airport|           Wales|POINT (-168.095 6...|
```