Assignment 7: Partitioning

In [96]:
```
import pandas as pd
import numpy as np
import math

import hashlib
```

7.1.a

Read Dataset

In [48]:
```
flight_df = pd.read_parquet('routes.parquet')
```

In [49]:
```
flight_df.head()
```

Out[49]:

	airline	src_airport	dst_airport	codeshare	equipment
0	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2965.0, 'name': 'Sochi Internat...	{'airport_id': 2990.0, 'name': 'Kazan Internat...	False	[CR2]
1	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2966.0, 'name': 'Astrakhan Airp...	{'airport_id': 2990.0, 'name': 'Kazan Internat...	False	[CR2]
2	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2966.0, 'name': 'Astrakhan Airp...	{'airport_id': 2962.0, 'name': 'Mineralnyye Vo...	False	[CR2]
3	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2968.0, 'name': 'Chelyabinsk Ba...	{'airport_id': 2990.0, 'name': 'Kazan Internat...	False	[CR2]
4	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2968.0, 'name': 'Chelyabinsk Ba...	{'airport_id': 4078.0, 'name': 'Tolmachevo Air...	False	[CR2]

Add Key Column

In [62]:
```
flight_df['key'] = flight_df.apply(lambda _: '', axis=1)

keys = []

for i, row in flight_df.iterrows():
    try:
        s = row['src_airport']['iata']
        d = row['dst_airport']['iata']
        a = row['airline']['iata']
        key = s + d + a
        keys.append(key)
        #row['key'] = str(key)
    except:
```

```
        keys.append(np.nan)
        pass

flight_df['key'] = keys
```

In [64]:
```
flight_df = flight_df.dropna(subset='key')
```

In [65]:
```
flight_df.head()
```

Out[65]:

	airline	src_airport	dst_airport	codeshare	equipment	key	kv_key
0	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2965.0, 'name': 'Sochi Internat...	{'airport_id': 2990.0, 'name': 'Kazan Internat...	False	[CR2]	AERKZN2B	A
1	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2966.0, 'name': 'Astrakhan Airp...	{'airport_id': 2990.0, 'name': 'Kazan Internat...	False	[CR2]	ASFKZN2B	A
2	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2966.0, 'name': 'Astrakhan Airp...	{'airport_id': 2962.0, 'name': 'Mineralnyye Vo...	False	[CR2]	ASFMRV2B	A
3	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2968.0, 'name': 'Chelyabinsk Ba...	{'airport_id': 2990.0, 'name': 'Kazan Internat...	False	[CR2]	CEKKZN2B	C
4	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2968.0, 'name': 'Chelyabinsk Ba...	{'airport_id': 4078.0, 'name': 'Tolmachevo Air...	False	[CR2]	CEKOVB2B	C

Partitioning the Data

In [52]:
```
partitions = (
        ('A', 'A'), ('B', 'B'), ('C', 'D'), ('E', 'F'),
        ('G', 'H'), ('I', 'J'), ('K', 'L'), ('M', 'M'),
        ('N', 'N'), ('O', 'P'), ('Q', 'R'), ('S', 'T'),
        ('U', 'U'), ('V', 'V'), ('W', 'X'), ('Y', 'Z')
    )
```

In [74]:
```
flight_df['kv_key'] = flight_df['key'].str[:1]
```

In [75]:
```
keys2 = []

for i, row in flight_df.iterrows():
    if row['kv_key'] in ['A', 'B', 'M', 'N', 'U', 'V']:
        keys2.append(row['kv_key'])
    elif row['kv_key'] in ['C', 'D']:
```

```
        keys2.append('C-D')
    elif row['kv_key'] in ['E', 'F']:
        keys2.append('E-F')
    elif row['kv_key'] in ['G', 'H']:
        keys2.append('G-H')
    elif row['kv_key'] in ['I', 'J']:
        keys2.append('I-J')
    elif row['kv_key'] in ['K', 'L']:
        keys2.append('K-L')
    elif row['kv_key'] in ['O', 'P']:
        keys2.append('O-P')
    elif row['kv_key'] in ['Q', 'R']:
        keys2.append('Q-R')
    elif row['kv_key'] in ['S', 'T']:
        keys2.append('S-T')
    elif row['kv_key'] in ['W', 'X']:
        keys2.append('W-X')
    elif row['kv_key'] in ['Y', 'Z']:
        keys2.append('Y-Z')
```

In [76]:
```
flight_df['kv_key'] = keys2
```

In [77]:
```
partition_cols = ['kv_key']
```

In [78]:
```
flight_df.to_parquet('results/kv', partition_cols = partition_cols)
```

7.1.b

Hash Keys

In [80]:
```
def hash_key(key):
    m = hashlib.sha256()
    m.update(str(key).encode('utf-8'))
    return m.hexdigest()
```

In [90]:
```
flight_df['hashed'] = flight_df.key.apply(hash_key)

flight_df['hash_key'] = flight_df['hashed'].str[0]
flight_df['hash_key'] = flight_df['hash_key'].str.upper()
```

In [91]:
```
flight_df.head()
```

Out[91]:

	airline	src_airport	dst_airport	codeshare	equipment	key	kv_key	
0	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2965.0, 'name': 'Sochi Internat...	{'airport_id': 2990.0, 'name': 'Kazan Internat...	False	[CR2]	AERKZN2B	A	652(
1	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2966.0, 'name': 'Astrakhan Airp...	{'airport_id': 2990.0, 'name': 'Kazan Internat...	False	[CR2]	ASFKZN2B	A	9eea5
2	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2966.0, 'name': 'Astrakhan Airp...	{'airport_id': 2962.0, 'name': 'Mineralnyye Vo...	False	[CR2]	ASFMRV2B	A	1611
3	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2968.0, 'name': 'Chelyabinsk Ba...	{'airport_id': 2990.0, 'name': 'Kazan Internat...	False	[CR2]	CEKKZN2B	C-D	39aa9
4	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2968.0, 'name': 'Chelyabinsk Ba...	{'airport_id': 4078.0, 'name': 'Tolmachevo Air...	False	[CR2]	CEKOVB2B	C-D	143b

Partitioning the Data

In [92]:
```
partition_cols = ['hash_key']
```

In [94]:
```
flight_df.to_parquet('results/hash', partition_cols = partition_cols)
```

7.1.c

Extracting Locations

In [100…
```
coords = []

for i, row in flight_df.iterrows():
    coord = []
    lat = row['src_airport']['latitude']
    lon = row['src_airport']['longitude']
    coord.append(lat)
    coord.append(lon)
    coords.append(coord)
```

In [101…
```
flight_df['coordinates'] = coords
```

Determining Location

In [95]:
```
west = [45.5945645, -121.1786823]
central = [41.1544433, -96.0422378]
east = [39.08344, -77.6497145]
```

In [105…
```
def nearest_center(loc):
    w = math.dist(loc, west)
    c = math.dist(loc, central)
    e = math.dist(loc, east)
    data = {'west': w, 'central': c, 'east': e}
    closest = min(data, key=data.get)
    return closest
```

In [106…
```
flight_df['location'] = flight_df.coordinates.apply(nearest_center)
```

In [110…
```
flight_df.head()
```

Out[110]:

	airline	src_airport	dst_airport	codeshare	equipment	key	kv_key	
0	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2965.0, 'name': 'Sochi Internat...	{'airport_id': 2990.0, 'name': 'Kazan Internat...	False	[CR2]	AERKZN2B	A	652
1	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2966.0, 'name': 'Astrakhan Airp...	{'airport_id': 2990.0, 'name': 'Kazan Internat...	False	[CR2]	ASFKZN2B	A	9eea
2	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2966.0, 'name': 'Astrakhan Airp...	{'airport_id': 2962.0, 'name': 'Mineralnyye Vo...	False	[CR2]	ASFMRV2B	A	161
3	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2968.0, 'name': 'Chelyabinsk Ba...	{'airport_id': 2990.0, 'name': 'Kazan Internat...	False	[CR2]	CEKKZN2B	C-D	39aa
4	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2968.0, 'name': 'Chelyabinsk Ba...	{'airport_id': 4078.0, 'name': 'Tolmachevo Air...	False	[CR2]	CEKOVB2B	C-D	143

Partitioning the Data

In [108…
```
partition_cols = ['location']
```

In [109…
```
flight_df.to_parquet('results/geo', partition_cols = partition_cols)
```

7.1.d

Defining the Function

In [133…
```
def balance_partitions(keys, num_partitions):
    partitions = []
    k = len(keys)/num_partitions
    for i, row in keys.iterrows():
        p = i // k
        partitions.append(p)
    return(partitions)
```

Example: Split Into 23 Partitions

In [142…
```
part_nums = balance_partitions(flight_df, 23)
```

In [145…
```
flight_df['partition'] = part_nums
flight_df['partition'] = flight_df['partition'].astype(int)
```

In [146…
```
flight_df.head()
```

Out[146]:

	airline	src_airport	dst_airport	codeshare	equipment	key	kv_key	
0	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2965.0, 'name': 'Sochi Internat...	{'airport_id': 2990.0, 'name': 'Kazan Internat...	False	[CR2]	AERKZN2B	A	65
1	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2966.0, 'name': 'Astrakhan Airp...	{'airport_id': 2990.0, 'name': 'Kazan Internat...	False	[CR2]	ASFKZN2B	A	9ee
2	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2966.0, 'name': 'Astrakhan Airp...	{'airport_id': 2962.0, 'name': 'Mineralnyye Vo...	False	[CR2]	ASFMRV2B	A	161
3	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2968.0, 'name': 'Chelyabinsk Ba...	{'airport_id': 2990.0, 'name': 'Kazan Internat...	False	[CR2]	CEKKZN2B	C-D	39aa
4	{'airline_id': 410, 'name': 'Aerocondor', 'ali...	{'airport_id': 2968.0, 'name': 'Chelyabinsk Ba...	{'airport_id': 4078.0, 'name': 'Tolmachevo Air...	False	[CR2]	CEKOVB2B	C-D	143

Partitioning the Data

In [147…
```
partition_cols = ['partition']
```

In [149…

```
flight_df.to_parquet('results/partition', partition_cols = partition_cols)
```