In [1]:

```
## Import pandas and data

import pandas as pd
```

In [7]:

```
airlines=pd.read_csv('/Users/vyshnavigovindankutty/Documents/Project_1_Airport_data/Airport_data/airlines.csv',sep=',');
airport_code=pd.read_csv('/Users/vyshnavigovindankutty/Documents/Project_1_Airport_data/Airport_data/airportcode.csv',sep=',');
```

In [8]:

```
airlines.head()
```

Out[8]:

	Year	Month	DayofMonth	DayOfWeek	DepTime	CRSDepTime	ArrTime	CRSArrTime	Uni
0	2008	1	16	3	1725.0	1735	1959.0	2021	
1	2008	1	17	4	1717.0	1701	1915.0	1855	
2	2008	1	17	4	1220.0	1225	1440.0	1504	
3	2008	1	17	4	1530.0	1530	1645.0	1637	
4	2008	1	17	4	1203.0	1205	1429.0	1429	

5 rows × 31 columns

In [9]:

```
airport_code.head()
```

Out[9]:

	City	State	Country	IATA
0	Abbotsford	BC	Canada	YXX
1	Aberdeen	SD	USA	ABR
2	Abilene	TX	USA	ABI
3	Akron	OH	USA	CAK
4	Alamosa	CO	USA	ALS

In [ ]:

```
### ANALYSIS QUERIES
```

In [3]:

```
##1 Count of flights that departed late at origin and reached their destination early or on time

len(airlines[(airlines.IsDepDelayed=='YES')&(airlines.IsArrDelayed=='NO')])
```

Out[3]:

```
54233
```

In [6]:

```
##2 Count of flights which departed late from origin by more than 60 minutes

len(airlines[((airlines.IsDepDelayed=='YES')&((airlines['DepTime'].apply(pd.to_numeric,errors='coerce'))>airlines.CRSDepTime+100))|((airlines.IsDepDelayed=='YES')&((airlines['DepTime'].apply(pd.to_numeric,errors='coerce'))<airlines.CRSDepTime)&(((2400+(airlines['DepTime'].apply(pd.to_numeric,errors='coerce')))-airlines.CRSDepTime)>100))])
```

Out[6]:

```
40104
```

In [7]:

```
##3 Count of flights which departed early or on time but arrived late by at least 15 minutes

len(airlines[((airlines.IsDepDelayed=='YES')&((airlines['DepTime'].apply(pd.to_numeric,errors='coerce'))>airlines.CRSDepTime+15))|((airlines.IsDepDelayed=='YES')&((airlines['DepTime'].apply(pd.to_numeric,errors='coerce'))<airlines.CRSDepTime)&(((2400+(airlines['DepTime'].apply(pd.to_numeric,errors='coerce')))-airlines.CRSDepTime)>15))])
```

Out[7]:

```
132792
```

In [8]:

```
##4 Count of flights departed from following major airports - ORD, DFW, ATL, LAX, SFO

len(airlines[airlines['Origin'].isin(['ORD','DFW','ATL','LAX','SFO'])])
```

Out[8]:

```
118212
```

In [9]:

```
##5 Add a column FlightDate by using Year, Month and DayOfMonth. Format should be yyyyMMdd

airlines['FlightDate']=airlines['Year'].astype(str).str.cat(airlines['Month'].astype(str).apply(lambda x:x.zfill(2))).str.cat(airlines['DayofMonth'].astype(str).apply(lambda x:x.zfill(2)));
```

In [10]:

```
##6 Count of flights that departed late between January 1 2008 to January 9 2008 using FlightDate

len(airlines[(airlines.IsDepDelayed=='YES')&(pd.to_datetime(airlines['FlightDate'])>'20080101')&(pd.to_datetime(airlines['FlightDate'])<'20080109')])
```

Out[10]:

```
73653
```

In [11]:

```
##7 Count of flights that departed late on Sundays

len(airlines[(airlines.IsDepDelayed=='YES')&(airlines['DayOfWeek']==7)])
```

Out[11]:

```
34708
```

In [ ]:

```
##8 Get number of flights that had delayed departure and number of flights delayed in arrival for each day along with number of flights departed for each day for January 2009
        #i. Output should contain 4 columns - FlightDate, FlightCount, DepDelayedCount, ArrDelayedCount
        #ii.FlightDate should be of YYYY-MM-dd format.
        #iii. Data should be sorted in ascending order by flightDate
```

In [13]:

```
##9 Get number of airports (IATA Codes) for each state in the US. Sort the data in descending order by count

airport_code[airport_code.Country=='USA'].groupby('State').size().to_frame('Count').sort_values('Count',ascending=False)
```

Out[13]:

State	Count
CA	29
TX	26
AK	25
NY	18
FL	18
MI	18
MT	14
PA	13
IL	12

CO	12
NC	10
WY	10
NE	9
WI	9
KS	9
WA	9
GA	9
NM	9
HI	9
MN	8
ND	8
AZ	8
MO	8
IA	8
AR	8
MA	8
WV	8
VA	7
OR	7
SD	7
MS	7
ME	7
LA	7
AL	6
IN	6
TN	6
SC	6
ID	6
OH	6
OK	5
KY	4

NH	3
MD	3
VT	3
NV	3
NJ	3
UT	2
CT	2
Hawaii	2
DE	1
RI	1

In [14]:
```
##10 Get number of flights departed from each US airport

airlines.merge(airport_code[airport_code.Country=='USA'],left_on='Orig
in',right_on='IATA',how='inner').groupby('Origin').size()
```

Out[14]:
```
Origin
ABE      413
ABI      240
ABQ     3447
ABY      102
ACT      209
        ... 
WRG       62
XNA     1199
YAK       62
YKM       33
YUM      380
Length: 270, dtype: int64
```

In [15]:
```
##11 Get number of flights departed from each US state

airlines.merge(airport_code[airport_code.Country=='USA'],left_on='Orig
in',right_on='IATA',how='inner').groupby('State').size().head()
```

Out[15]:
```
State
AK     2818
AL     3931
AR     2928
AZ    20768
CA    72853
dtype: int64
```

In [16]:

```
##12 Get the list of airports in the US from which flights have not departed

airport_code[~(airport_code['IATA'].isin(airlines['Origin']))&(airport_code['Country']=='USA')][['IATA','Country']].sort_values('IATA',ascending=True).head(10)
```

Out[16]:

	IATA	Country
1	ABR	USA
322	ACK	USA
20	AHN	USA
10	AIA	USA
242	AKN	USA
4	ALS	USA
496	ALW	USA
12	AOO	USA
323	APF	USA
11	APN	USA

In [17]:

```
##13 Check if there are any origins in airlines data which do not have record in airport-codes

airlines[~(airlines['Origin'].isin(airport_code['IATA']))].Origin.unique()
```

Out[17]:

```
array(['HDN', 'SJU', 'ITO', 'KOA', 'STT', 'OTZ', 'BQN', 'STX', 'PMD',
       'CEC', 'PSE', 'SCC', 'SLE', 'CDC', 'PSG', 'ADK'], dtype=object)
```

In [18]:

```
##14 Get the total number of flights from the airports that do not contain entries in airport-codes

len(airlines[~(airlines['Origin'].isin(airport_code['IATA']))])
```

Out[18]:

```
5585
```

In [19]:

```
##15 Get the total number of flights per airport that do not contain entries in airport-codes

airlines[~(airlines['Origin'].isin(airport_code['IATA']))].groupby('Origin').size()
```

Out[19]:

```
Origin
ADK       9
BQN     124
CDC      48
CEC      88
HDN     429
ITO     786
KOA    1316
OTZ      92
PMD      57
PSE     110
PSG      62
SCC      62
SJU    1997
SLE      54
STT     311
STX      40
dtype: int64
```

In [ ]:

```
```

In [ ]:

```
```