GROUP NAMES:

- Gilbert Andze Tsoungui 211262656 christ18
- Jaskaran Garha 214314439 garhajas
- David Geller 214404255 dgeller
- William Zhen 214305171 will636

This project has been done with jupyter notebook

Data Source

[Data Source Link (https://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236)](https://www.transtats.bts.gov/DL_SelectFields.asp?Table_ID=236)

Data Ingestion

Our Plan in this project was to use spark to process CSV files coming from the Bureau of Transportation Statistics and use graphframe to load data in which airport are represented as vertices and flight as edges. We have encountered some problems in that process.

- ##### We quickly realized that spark has a lot of dependencies, some of our functions didn't work because we didn't have the right scala version install or the right python version install.
- ##### it 's was easier for us to move to panda and Jupyter because panda only require python

In [10]:
```
import pandas as pd
```

2017 data analysis

In [11]:
```
data_2017 = pd.read_csv('2017.csv')
```

In [12]:
```
data_2017.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 464205 entries, 0 to 464204
Data columns (total 40 columns):
YEAR                         464205 non-null int64
QUARTER                      464205 non-null int64
MONTH                        464205 non-null int64
DAY_OF_MONTH                 464205 non-null int64
DAY_OF_WEEK                  464205 non-null int64
FL_DATE                      464205 non-null object
OP_UNIQUE_CARRIER            464205 non-null object
OP_CARRIER_AIRLINE_ID        464205 non-null int64
OP_CARRIER                   464205 non-null object
ORIGIN_AIRPORT_ID            464205 non-null int64
ORIGIN_AIRPORT_SEQ_ID        464205 non-null int64
ORIGIN_CITY_MARKET_ID        464205 non-null int64
ORIGIN                       464205 non-null object
ORIGIN_CITY_NAME             464205 non-null object
ORIGIN_STATE_ABR             464205 non-null object
ORIGIN_STATE_FIPS            464205 non-null int64
ORIGIN_STATE_NM              464205 non-null object
ORIGIN_WAC                   464205 non-null int64
DEST_AIRPORT_ID              464205 non-null int64
DEST_AIRPORT_SEQ_ID          464205 non-null int64
DEST_CITY_MARKET_ID          464205 non-null int64
DEST                         464205 non-null object
DEST_CITY_NAME               464205 non-null object
DEST_STATE_ABR               464205 non-null object
DEST_STATE_FIPS              464205 non-null int64
DEST_STATE_NM                464205 non-null object
DEST_WAC                     464205 non-null int64
DEP_DELAY                    459063 non-null float64
DEP_DELAY_NEW                459063 non-null float64
ARR_TIME                     458665 non-null float64
ARR_DELAY                    457892 non-null float64
ARR_DELAY_NEW                457892 non-null float64
CANCELLED                    464205 non-null float64
CANCELLATION_CODE            5324 non-null object
CARRIER_DELAY                85302 non-null float64
WEATHER_DELAY                85302 non-null float64
NAS_DELAY                    85302 non-null float64
SECURITY_DELAY               85302 non-null float64
LATE_AIRCRAFT_DELAY          85302 non-null float64
Unnamed: 39                  0 non-null float64
dtypes: float64(12), int64(16), object(12)
memory usage: 141.7+ MB
```

In [13]:
```
data_2017.describe()
```
Out[13]:

	YEAR	QUARTER	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	OP_CARRIER_AIRLINE_ID
count	464205.0	464205.0	464205.0	464205.000000	464205.000000	464205.000000
mean	2017.0	4.0	12.0	16.056678	4.126043	19896.346211
std	0.0	0.0	0.0	8.890807	1.980423	382.736157
min	2017.0	4.0	12.0	1.000000	1.000000	19393.000000
25%	2017.0	4.0	12.0	8.000000	2.000000	19690.000000
50%	2017.0	4.0	12.0	16.000000	4.000000	19805.000000
75%	2017.0	4.0	12.0	24.000000	6.000000	20304.000000
max	2017.0	4.0	12.0	31.000000	7.000000	21171.000000

8 rows × 28 columns

In [14]:
```
data_2017.head()
```
Out[14]:

	YEAR	QUARTER	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	FL_DATE	OP_UNIQUE_CARRIER
0	2017	4	12	1	5	2017-12-01	OO
1	2017	4	12	1	5	2017-12-01	OO
2	2017	4	12	1	5	2017-12-01	OO
3	2017	4	12	1	5	2017-12-01	OO
4	2017	4	12	1	5	2017-12-01	OO

5 rows × 40 columns

Data Cleaning

As We can see in the above schema We don't need to do any cleaning on the data

Questions to answer (2017)

Which Trip have the most delays and cancellations?

In [15]:
```
trip_with_delay_2017 = data_2017.groupby(["ORIGIN_CITY_NAME","DEST_CITY_
NAME"])["ARR_DELAY"].mean()\
.sort_values(ascending=False).head(10)
```

In [16]:
```
trip_with_delay_2017
```

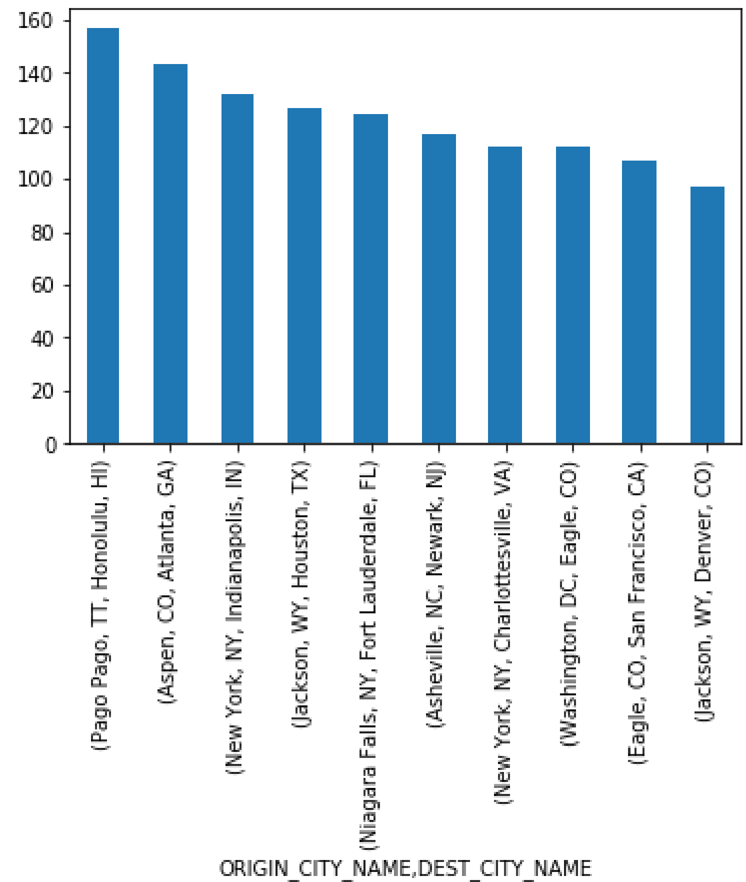
Out[16]:
```
ORIGIN_CITY_NAME   DEST_CITY_NAME
Pago Pago, TT      Honolulu, HI            156.600000
Aspen, CO          Atlanta, GA             143.400000
New York, NY       Indianapolis, IN        132.000000
Jackson, WY        Houston, TX             126.750000
Niagara Falls, NY  Fort Lauderdale, FL     124.500000
Asheville, NC      Newark, NJ              116.741935
New York, NY       Charlottesville, VA     112.500000
Washington, DC     Eagle, CO               112.500000
Eagle, CO          San Francisco, CA       107.000000
Jackson, WY        Denver, CO               97.307692
Name: ARR_DELAY, dtype: float64
```

Vizualization With matplotlib

In [17]:
```
%matplotlib inline
```

In [18]:
```
trip_with_delay_2017.plot(kind='bar', x='ORIGIN_CITY_NAME', y='ARR_DELAY')
```

Out[18]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x11b1b5fd0>
```



Which Airline companies have the most cancellations ?

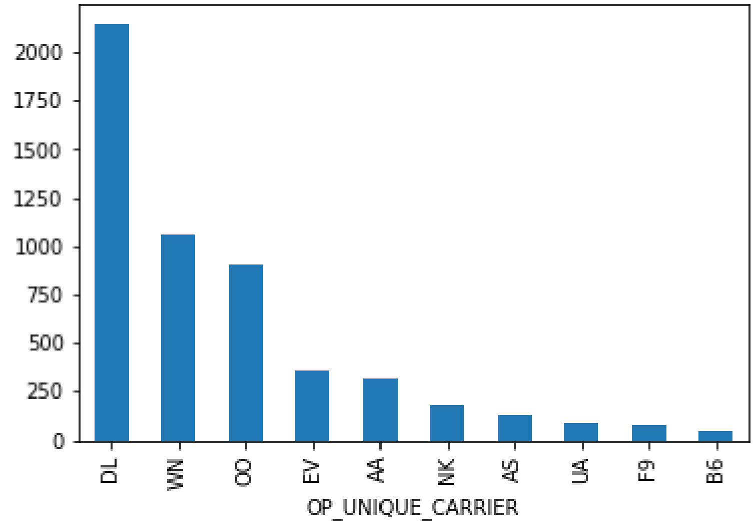
In [19]:
```
airline_with_most_cancellations_2017 = data_2017.groupby("OP_UNIQUE_CARRIER")["CANCELLATION_CODE"].apply(lambda x: x.notnull().sum())\
.sort_values(ascending=False).head(10)
```

In [20]:
```
airline_with_most_cancellations_2017
```

Out[20]:
```
OP_UNIQUE_CARRIER
DL    2141
WN    1061
OO     901
EV     356
AA     312
NK     179
AS     125
UA      84
F9      77
B6      44
Name: CANCELLATION_CODE, dtype: int64
```

In [21]:
```
airline_with_most_cancellations_2017.plot(kind='bar', x='OP_UNIQUE_CARRIER')
```

Out[21]: <matplotlib.axes._subplots.AxesSubplot at 0x11b7c07d0>



most common cause of delays

In [22]:
```
weather_2017_delays_count = data_2017[data_2017['WEATHER_DELAY'] > 0.0]["WEATHER_DELAY"].notnull().sum()
```

In [23]:
```
weather_2017_delays_count
```

Out[23]: 4913

In [24]:
```
security_2017_delays_count = data_2017[data_2017['SECURITY_DELAY'] > 0.0]['SECURITY_DELAY'].notnull().sum()
```

In [25]:
```
security_2017_delays_count
```

Out[25]: 417

In [27]:
```
late_aircraft_2017_delays_count = data_2017[data_2017['LATE_AIRCRAFT_DELAY'] > 0.0]['LATE_AIRCRAFT_DELAY'].notnull().sum()
```

In [28]:
```
late_aircraft_2017_delays_count
```

Out[28]: 44845

In [31]:
```
carrier_2017_delays_count = data_2017[data_2017['CARRIER_DELAY'] > 0.0]['CARRIER_DELAY'].notnull().sum()
```

In [32]:
```
carrier_2017_delays_count
```
Out[32]: 46455

In [33]:
```
nas_2017_delays_count = data_2017[data_2017['NAS_DELAY'] > 0.0]['NAS_DEL
AY'].notnull().sum()
```

In [34]:
```
nas_2017_delays_count
```
Out[34]: 44392

Now We need to create a new dataframe with the data we have and then vizualize it

In [41]:
```
delay_causes_data_2017 = {
    "Delays Causes":  ["SECURITY_DELAY","WEATHER_DELAY","LATE_AIRCRAFT_D
ELAY","CARRIER_DELAY","NAS_DELAY"],
    "Values": [security_2017_delays_count,weather_2017_delays_count,late
_aircraft_2017_delays_count,
              carrier_2017_delays_count,nas_2017_delays_count]

}
```

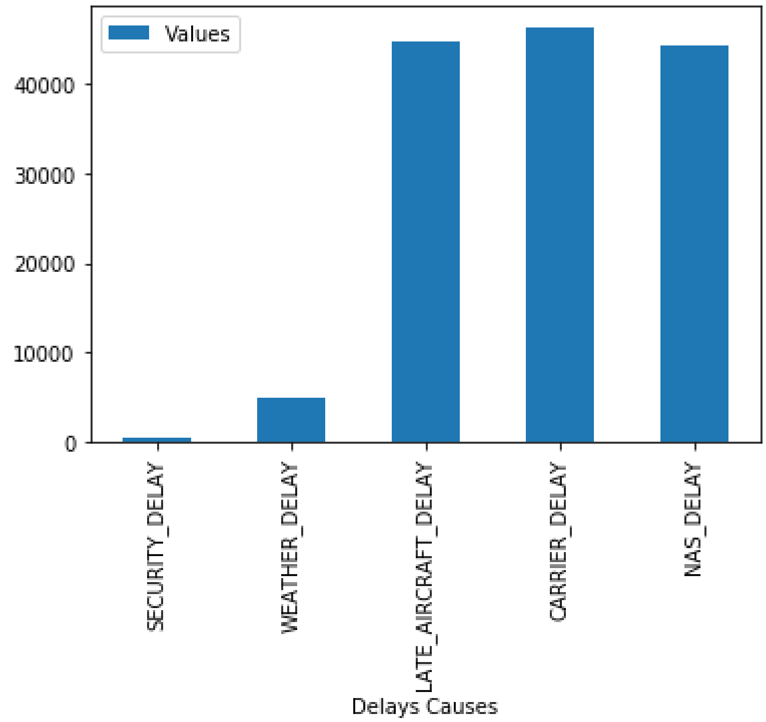
In [42]:
```
df_delays_causes_2017 = pd.DataFrame(delay_causes_data_2017,
                                index=["SECURITY_DELAY","WEATHER_DELAY",
"LATE_AIRCRAFT_DELAY","CARRIER_DELAY","NAS_DELAY"])
```

In [43]:
```
df_delays_causes_2017
```
Out[43]:

	Delays Causes	Values
SECURITY_DELAY	SECURITY_DELAY	417
WEATHER_DELAY	WEATHER_DELAY	4913
LATE_AIRCRAFT_DELAY	LATE_AIRCRAFT_DELAY	44845
CARRIER_DELAY	CARRIER_DELAY	46455
NAS_DELAY	NAS_DELAY	44392

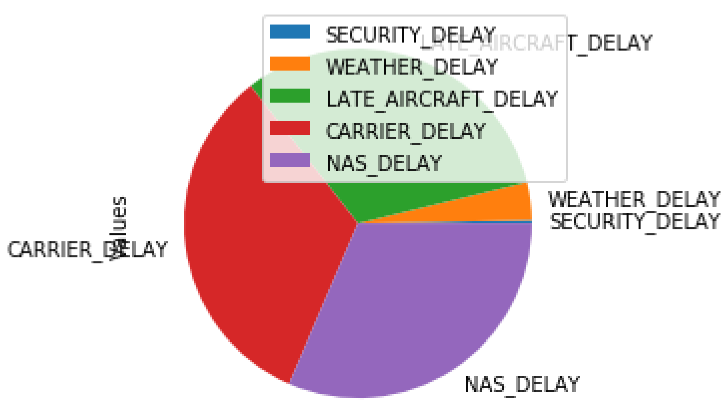
In [44]:
```
df_delays_causes_2017.plot(kind='bar',x="Delays Causes", y="Values")
```

Out[44]: <matplotlib.axes._subplots.AxesSubplot at 0x11aa85350>



In [45]:
```
df_delays_causes_2017.plot(kind='pie',x="Delays Causes", y="Values")
```

Out[45]: <matplotlib.axes._subplots.AxesSubplot at 0x11ab41490>



Now We need To repeat the same process for 2018 and 2019 data and then we will summarized our finding

2018 data analysis

In [48]:
```
data_2018 = pd.read_csv('2018.csv')
```

In [49]:
```
data_2018.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 593842 entries, 0 to 593841
Data columns (total 40 columns):
YEAR                        593842 non-null int64
QUARTER                     593842 non-null int64
MONTH                       593842 non-null int64
DAY_OF_MONTH                593842 non-null int64
DAY_OF_WEEK                 593842 non-null int64
FL_DATE                     593842 non-null object
OP_UNIQUE_CARRIER           593842 non-null object
OP_CARRIER_AIRLINE_ID       593842 non-null int64
OP_CARRIER                  593842 non-null object
ORIGIN_AIRPORT_ID           593842 non-null int64
ORIGIN_AIRPORT_SEQ_ID       593842 non-null int64
ORIGIN_CITY_MARKET_ID       593842 non-null int64
ORIGIN                      593842 non-null object
ORIGIN_CITY_NAME            593842 non-null object
ORIGIN_STATE_ABR            593842 non-null object
ORIGIN_STATE_FIPS           593842 non-null int64
ORIGIN_STATE_NM             593842 non-null object
ORIGIN_WAC                  593842 non-null int64
DEST_AIRPORT_ID             593842 non-null int64
DEST_AIRPORT_SEQ_ID         593842 non-null int64
DEST_CITY_MARKET_ID         593842 non-null int64
DEST                        593842 non-null object
DEST_CITY_NAME              593842 non-null object
DEST_STATE_ABR              593842 non-null object
DEST_STATE_FIPS             593842 non-null int64
DEST_STATE_NM               593842 non-null object
DEST_WAC                    593842 non-null int64
DEP_DELAY                   587316 non-null float64
DEP_DELAY_NEW               587316 non-null float64
ARR_TIME                    586812 non-null float64
ARR_DELAY                   585737 non-null float64
ARR_DELAY_NEW               585737 non-null float64
CANCELLED                   593842 non-null float64
CANCELLATION_CODE           6752 non-null object
CARRIER_DELAY               108682 non-null float64
WEATHER_DELAY               108682 non-null float64
NAS_DELAY                   108682 non-null float64
SECURITY_DELAY              108682 non-null float64
LATE_AIRCRAFT_DELAY         108682 non-null float64
Unnamed: 39                 0 non-null float64
dtypes: float64(12), int64(16), object(12)
memory usage: 181.2+ MB
```

In [51]:
```
data_2018.describe()
```

Out[51]:

	YEAR	QUARTER	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	OP_CARRIER_AIRLINE_ID
count	593842.0	593842.0	593842.0	593842.000000	593842.000000	593842.000000
mean	2018.0	4.0	12.0	16.097708	4.068542	19982.960934
std	0.0	0.0	0.0	8.882057	2.056023	377.005582
min	2018.0	4.0	12.0	1.000000	1.000000	19393.000000
25%	2018.0	4.0	12.0	8.000000	2.000000	19790.000000
50%	2018.0	4.0	12.0	16.000000	4.000000	19977.000000
75%	2018.0	4.0	12.0	24.000000	6.000000	20368.000000
max	2018.0	4.0	12.0	31.000000	7.000000	20452.000000

8 rows × 28 columns

In [52]:
```
data_2018.head()
```

Out[52]:

	YEAR	QUARTER	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	FL_DATE	OP_UNIQUE_CARRIER
0	2018	4	12	6	4	2018-12-06	DL
1	2018	4	12	6	4	2018-12-06	DL
2	2018	4	12	6	4	2018-12-06	DL
3	2018	4	12	6	4	2018-12-06	DL
4	2018	4	12	6	4	2018-12-06	DL

5 rows × 40 columns

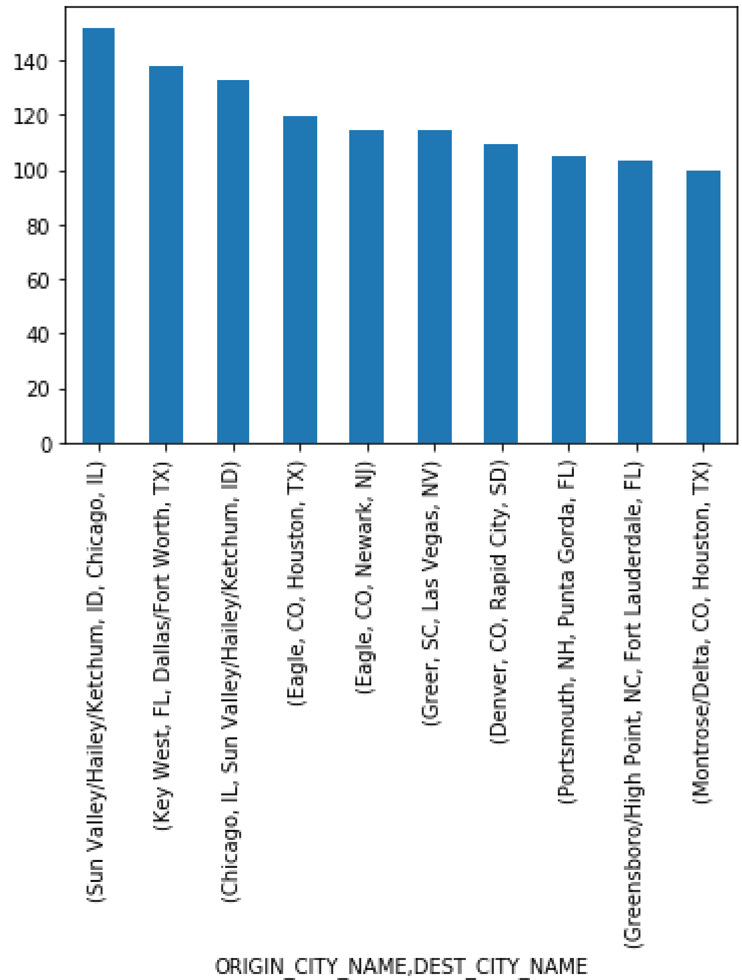
In [53]:
```
trip_with_delay_2018 = data_2018.groupby(["ORIGIN_CITY_NAME","DEST_CITY_NAME"])["ARR_DELAY"].mean()\
.sort_values(ascending=False).head(10)
```

In [54]:
```
trip_with_delay_2018
```

Out[54]:
```
ORIGIN_CITY_NAME                DEST_CITY_NAME
Sun Valley/Hailey/Ketchum, ID  Chicago, IL                        152.000
000
Key West, FL                   Dallas/Fort Worth, TX              138.250
000
Chicago, IL                    Sun Valley/Hailey/Ketchum, ID      133.000
000
Eagle, CO                      Houston, TX                        119.461
538
                               Newark, NJ                         114.666
667
Greer, SC                      Las Vegas, NV                      114.555
556
Denver, CO                     Rapid City, SD                     109.000
000
Portsmouth, NH                 Punta Gorda, FL                    105.090
909
Greensboro/High Point, NC      Fort Lauderdale, FL                103.555
556
Montrose/Delta, CO             Houston, TX                         99.692
308
Name: ARR_DELAY, dtype: float64
```

In [55]:
```
trip_with_delay_2018.plot(kind='bar', x='ORIGIN_CITY_NAME', y='ARR_DELA
Y')
```

Out[55]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x11b29ff90>
```



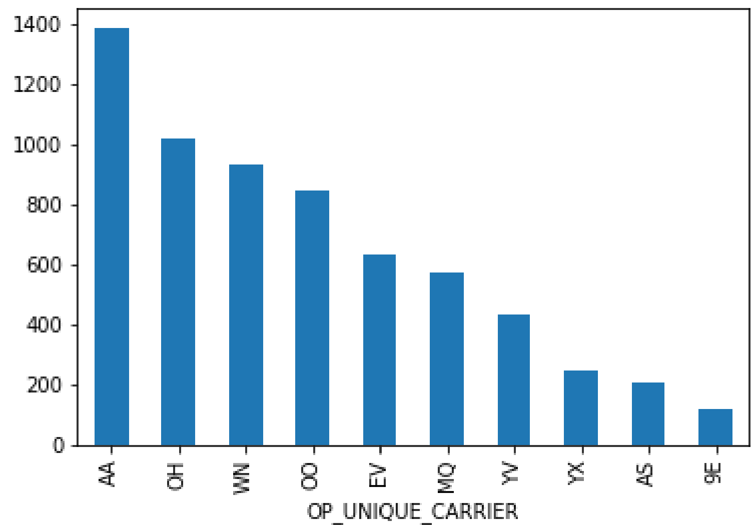
In [56]:
```
airline_with_most_cancellations_2018 = data_2018.groupby("OP_UNIQUE_CARR
IER")["CANCELLATION_CODE"].apply(lambda x: x.notnull().sum())\
.sort_values(ascending=False).head(10)
```

In [57]:
```
airline_with_most_cancellations_2018
```

Out[57]:
```
OP_UNIQUE_CARRIER
AA    1386
OH    1019
WN     931
OO     848
EV     630
MQ     572
YV     432
YX     243
AS     203
9E     116
Name: CANCELLATION_CODE, dtype: int64
```

In [58]:
```
airline_with_most_cancellations_2018.plot(kind='bar', x='OP_UNIQUE_CARRI
ER')
```

Out[58]: <matplotlib.axes._subplots.AxesSubplot at 0x122040490>



In [59]:
```
weather_2018_delays_count = data_2018[data_2018['WEATHER_DELAY'] > 0.0][
"WEATHER_DELAY"].notnull().sum()
```

In [60]:
```
weather_2018_delays_count
```

Out[60]: 5582

In [61]:
```
security_2018_delays_count = data_2018[data_2018['SECURITY_DELAY'] > 0.0
]['SECURITY_DELAY'].notnull().sum()
```

In [62]:
```
security_2018_delays_count
```

Out[62]: 504

In [63]:
```
late_aircraft_2018_delays_count = data_2018[data_2018['LATE_AIRCRAFT_DEL
AY'] > 0.0]['LATE_AIRCRAFT_DELAY'].notnull().sum()
```

In [64]:
```
late_aircraft_2018_delays_count
```

Out[64]: 56112

In [65]:
```
carrier_2018_delays_count = data_2018[data_2018['CARRIER_DELAY'] > 0.0][
'CARRIER_DELAY'].notnull().sum()
```

In [66]:
```
carrier_2018_delays_count
```

Out[66]: 55957

In [67]:
```
nas_2018_delays_count = data_2018[data_2018['NAS_DELAY'] > 0.0]['NAS_DEL
AY'].notnull().sum()
```

In [68]:
```
nas_2018_delays_count
```

Out[68]: 59514

In [69]:
```
delay_causes_data_2018 = {
    "Delays Causes":  ["SECURITY_DELAY","WEATHER_DELAY","LATE_AIRCRAFT_D
ELAY","CARRIER_DELAY","NAS_DELAY"],
    "Values": [security_2018_delays_count,weather_2018_delays_count,late
_aircraft_2018_delays_count,
              carrier_2018_delays_count,nas_2018_delays_count]

}
```

In [70]:
```
df_delays_causes_2018 = pd.DataFrame(delay_causes_data_2018,
                                 index=["SECURITY_DELAY","WEATHER_DELAY",
"LATE_AIRCRAFT_DELAY","CARRIER_DELAY","NAS_DELAY"])
```

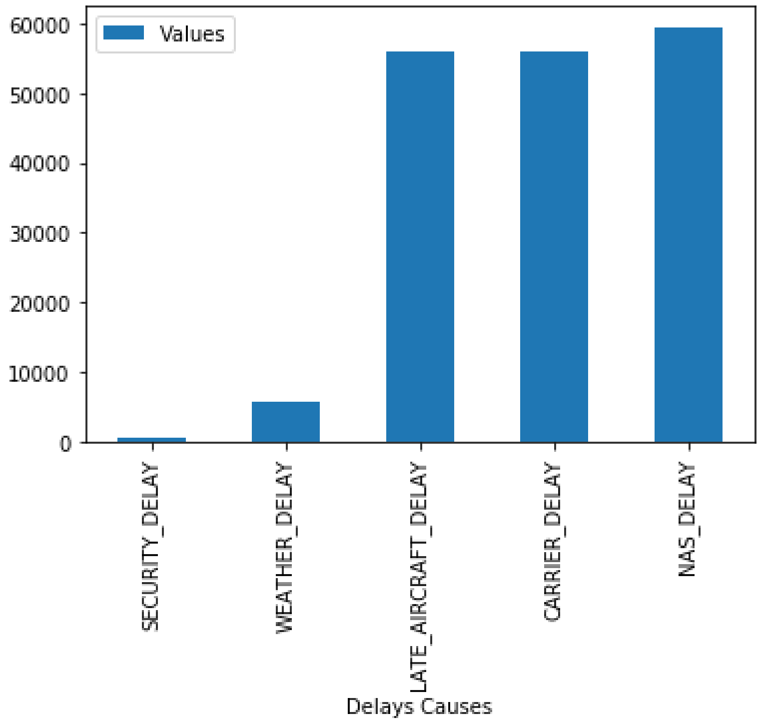
In [71]:
```
df_delays_causes_2018
```

Out[71]:

	Delays Causes	Values
SECURITY_DELAY	SECURITY_DELAY	504
WEATHER_DELAY	WEATHER_DELAY	5582
LATE_AIRCRAFT_DELAY	LATE_AIRCRAFT_DELAY	56112
CARRIER_DELAY	CARRIER_DELAY	55957
NAS_DELAY	NAS_DELAY	59514

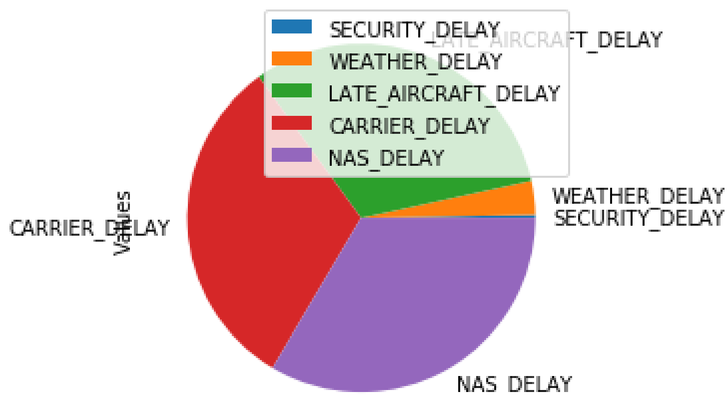
In [72]:
```
df_delays_causes_2018.plot(kind='bar',x="Delays Causes", y="Values")
```

Out[72]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x11eaf19d0>
```



In [73]:
```
df_delays_causes_2018.plot(kind='pie',x="Delays Causes", y="Values")
```

Out[73]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x120438650>
```



2019 data analysis

In [74]:
```
data_2019 = pd.read_csv('2019.csv')
```

In [75]:
```
data_2019.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 636014 entries, 0 to 636013
Data columns (total 40 columns):
YEAR                         636014 non-null int64
QUARTER                      636014 non-null int64
MONTH                        636014 non-null int64
DAY_OF_MONTH                 636014 non-null int64
DAY_OF_WEEK                  636014 non-null int64
FL_DATE                      636014 non-null object
OP_UNIQUE_CARRIER            636014 non-null object
OP_CARRIER_AIRLINE_ID        636014 non-null int64
OP_CARRIER                   636014 non-null object
ORIGIN_AIRPORT_ID            636014 non-null int64
ORIGIN_AIRPORT_SEQ_ID        636014 non-null int64
ORIGIN_CITY_MARKET_ID        636014 non-null int64
ORIGIN                       636014 non-null object
ORIGIN_CITY_NAME             636014 non-null object
ORIGIN_STATE_ABR             636014 non-null object
ORIGIN_STATE_FIPS            636014 non-null int64
ORIGIN_STATE_NM              636014 non-null object
ORIGIN_WAC                   636014 non-null int64
DEST_AIRPORT_ID              636014 non-null int64
DEST_AIRPORT_SEQ_ID          636014 non-null int64
DEST_CITY_MARKET_ID          636014 non-null int64
DEST                         636014 non-null object
DEST_CITY_NAME               636014 non-null object
DEST_STATE_ABR               636014 non-null object
DEST_STATE_FIPS              636014 non-null int64
DEST_STATE_NM                636014 non-null object
DEST_WAC                     636014 non-null int64
DEP_DELAY                    631100 non-null float64
DEP_DELAY_NEW                631100 non-null float64
ARR_TIME                     630680 non-null float64
ARR_DELAY                    629637 non-null float64
ARR_DELAY_NEW                629637 non-null float64
CANCELLED                    636014 non-null float64
CANCELLATION_CODE            5172 non-null object
CARRIER_DELAY                105046 non-null float64
WEATHER_DELAY                105046 non-null float64
NAS_DELAY                    105046 non-null float64
SECURITY_DELAY               105046 non-null float64
LATE_AIRCRAFT_DELAY          105046 non-null float64
Unnamed: 39                  0 non-null float64
dtypes: float64(12), int64(16), object(12)
memory usage: 194.1+ MB
```

In [76]:
```
data_2019.describe()
```

Out[76]:

	YEAR	QUARTER	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	OP_CARRIER_AIRLINE_ID
count	636014.0	636014.0	636014.0	636014.000000	636014.000000	636014.000000
mean	2019.0	4.0	10.0	15.870602	3.854994	19985.866814
std	0.0	0.0	0.0	8.884372	1.936610	374.079403
min	2019.0	4.0	10.0	1.000000	1.000000	19393.000000
25%	2019.0	4.0	10.0	8.000000	2.000000	19790.000000
50%	2019.0	4.0	10.0	16.000000	4.000000	19977.000000
75%	2019.0	4.0	10.0	24.000000	5.000000	20368.000000
max	2019.0	4.0	10.0	31.000000	7.000000	20452.000000

8 rows × 28 columns

In [77]:
```
data_2019.head()
```

Out[77]:

	YEAR	QUARTER	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	FL_DATE	OP_UNIQUE_CARRIER
0	2019	4	10	1	2	2019-10-01	AA
1	2019	4	10	2	3	2019-10-02	AA
2	2019	4	10	4	5	2019-10-04	AA
3	2019	4	10	5	6	2019-10-05	AA
4	2019	4	10	6	7	2019-10-06	AA

5 rows × 40 columns

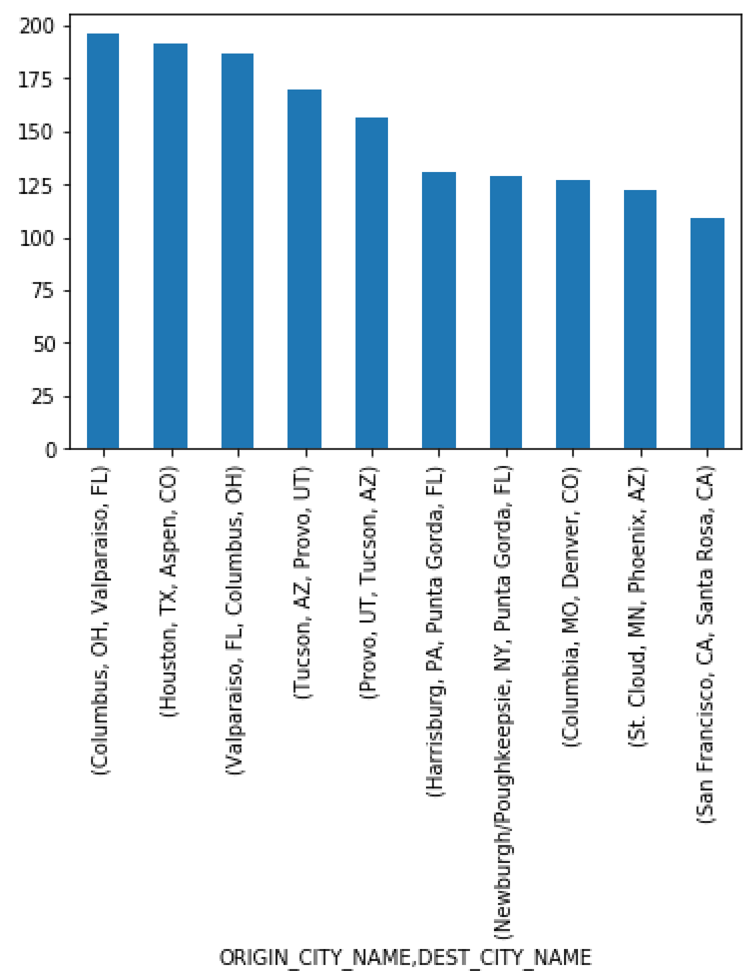
In [78]:
```
trip_with_delay_2019 = data_2019.groupby(["ORIGIN_CITY_NAME","DEST_CITY_NAME"])["ARR_DELAY"].mean()\
.sort_values(ascending=False).head(10)
```

In [79]:
```
trip_with_delay_2019
```

Out[79]:
```
ORIGIN_CITY_NAME            DEST_CITY_NAME
Columbus, OH                Valparaiso, FL     196.125000
Houston, TX                 Aspen, CO          191.750000
Valparaiso, FL              Columbus, OH       187.000000
Tucson, AZ                  Provo, UT          169.222222
Provo, UT                   Tucson, AZ         156.111111
Harrisburg, PA              Punta Gorda, FL    131.000000
Newburgh/Poughkeepsie, NY   Punta Gorda, FL    128.875000
Columbia, MO                Denver, CO         127.333333
St. Cloud, MN               Phoenix, AZ        122.250000
San Francisco, CA           Santa Rosa, CA     108.730769
Name: ARR_DELAY, dtype: float64
```

In [80]:
```
trip_with_delay_2019.plot(kind='bar', x='ORIGIN_CITY_NAME', y='ARR_DELAY')
```

Out[80]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x11a2d9910>
```



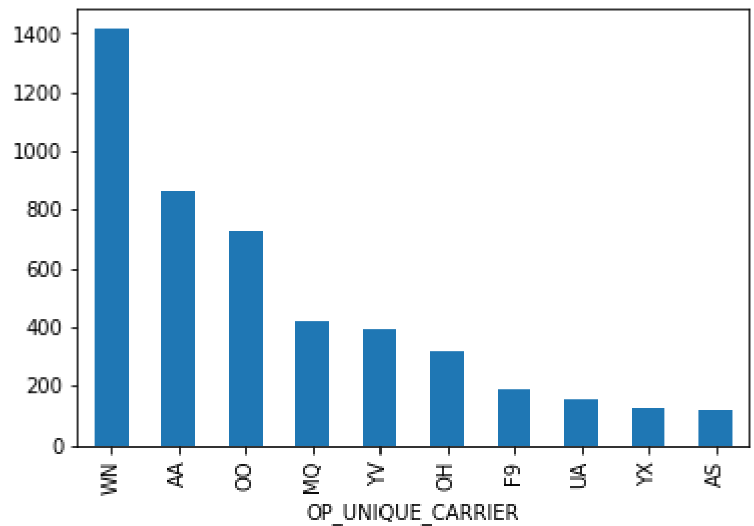
In [81]:
```
airline_with_most_cancellations_2019 = data_2019.groupby("OP_UNIQUE_CARRIER")["CANCELLATION_CODE"].apply(lambda x: x.notnull().sum())\
.sort_values(ascending=False).head(10)
```

In [82]:
```
airline_with_most_cancellations_2019
```

Out[82]:
```
OP_UNIQUE_CARRIER
WN    1414
AA     867
OO     725
MQ     420
YV     393
OH     316
F9     188
UA     151
YX     126
AS     121
Name: CANCELLATION_CODE, dtype: int64
```

In [83]:
```
airline_with_most_cancellations_2019.plot(kind='bar', x='OP_UNIQUE_CARRI
ER')
```

Out[83]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x127fb4f90>
```



In [84]:
```
weather_2019_delays_count = data_2019[data_2019['WEATHER_DELAY'] > 0.0][
"WEATHER_DELAY"].notnull().sum()
```

In [85]:
```
weather_2019_delays_count
```

Out[85]:
```
4292
```

In [86]:
```
security_2019_delays_count = data_2019[data_2019['SECURITY_DELAY'] > 0.0
]['SECURITY_DELAY'].notnull().sum()
```

In [87]:
```
security_2019_delays_count
```

Out[87]:
```
311
```

In [88]:
```
late_aircraft_2019_delays_count = data_2019[data_2019['LATE_AIRCRAFT_DEL
AY'] > 0.0]['LATE_AIRCRAFT_DELAY'].notnull().sum()
```

In [89]:
```
late_aircraft_2019_delays_count
```

Out[89]: 53000

In [90]:
```
carrier_2019_delays_count = data_2019[data_2019['CARRIER_DELAY'] > 0.0][
'CARRIER_DELAY'].notnull().sum()
```

In [91]:
```
carrier_2019_delays_count
```

Out[91]: 49049

In [92]:
```
nas_2019_delays_count = data_2019[data_2019['NAS_DELAY'] > 0.0]['NAS_DEL
AY'].notnull().sum()
```

In [93]:
```
nas_2019_delays_count
```

Out[93]: 58108

In [94]:
```
delay_causes_data_2019 = {
    "Delays Causes":  ["SECURITY_DELAY","WEATHER_DELAY","LATE_AIRCRAFT_D
ELAY","CARRIER_DELAY","NAS_DELAY"],
    "Values": [security_2019_delays_count,weather_2019_delays_count,late
_aircraft_2019_delays_count,
              carrier_2019_delays_count,nas_2019_delays_count]

}
```

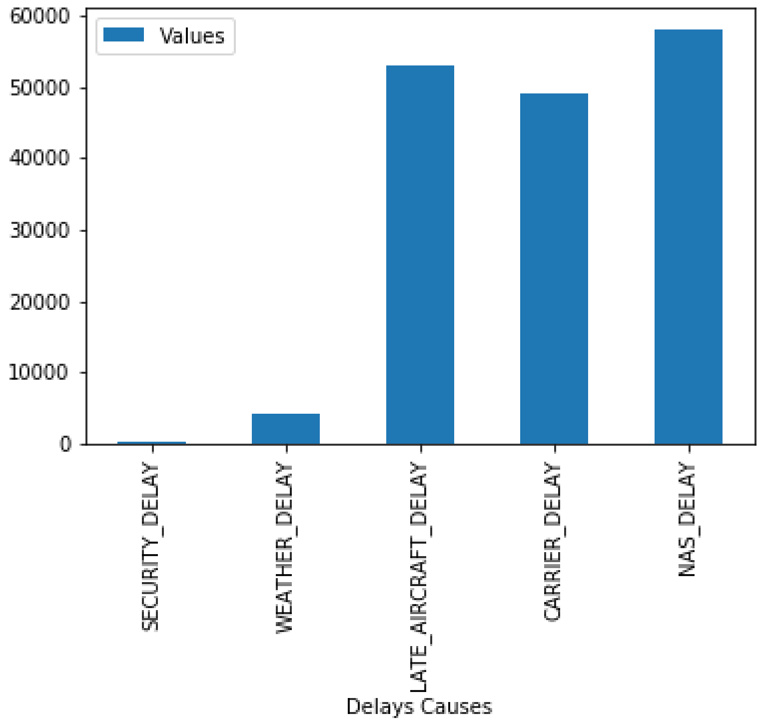
In [95]:
```
df_delays_causes_2019 = pd.DataFrame(delay_causes_data_2019,
                                    index=["SECURITY_DELAY","WEATHER_DELAY",
"LATE_AIRCRAFT_DELAY","CARRIER_DELAY","NAS_DELAY"])
```

In [96]:
```
df_delays_causes_2019
```

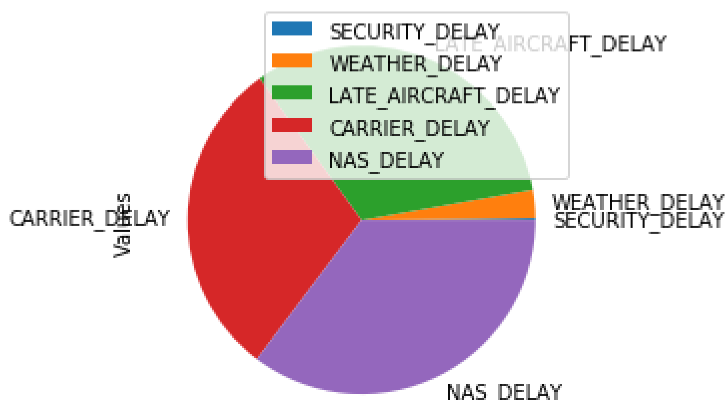
Out[96]:

	Delays Causes	Values
SECURITY_DELAY	SECURITY_DELAY	311
WEATHER_DELAY	WEATHER_DELAY	4292
LATE_AIRCRAFT_DELAY	LATE_AIRCRAFT_DELAY	53000
CARRIER_DELAY	CARRIER_DELAY	49049
NAS_DELAY	NAS_DELAY	58108

In [97]:
```
df_delays_causes_2019.plot(kind='bar',x="Delays Causes", y="Values")
```
Out[97]: <matplotlib.axes._subplots.AxesSubplot at 0x11a72fa50>



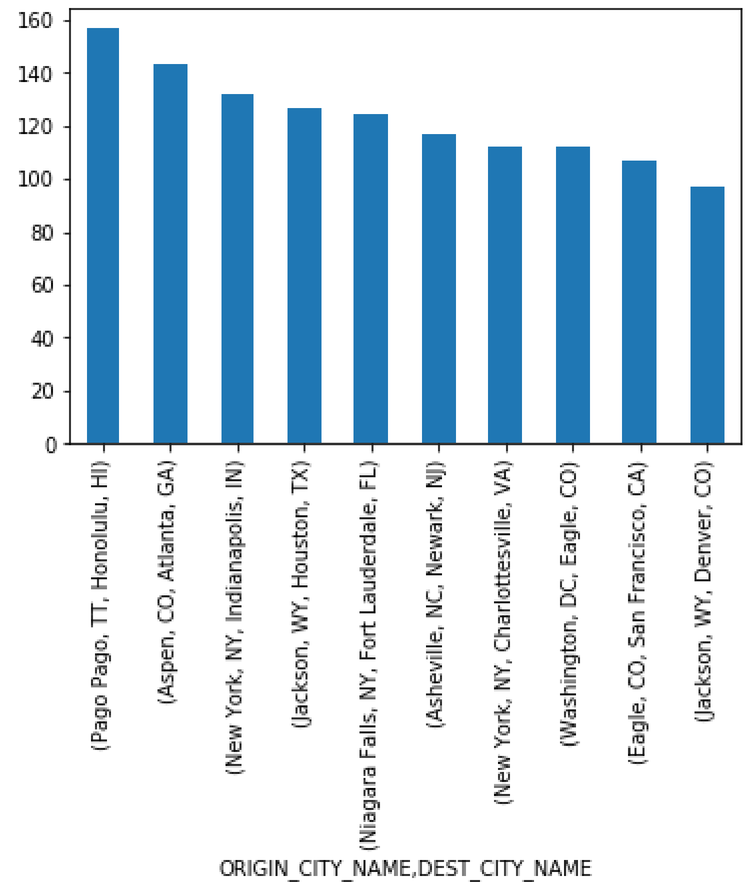
In [98]:
```
df_delays_causes_2019.plot(kind='pie',x="Delays Causes", y="Values")
```
Out[98]: <matplotlib.axes._subplots.AxesSubplot at 0x11a7cfa10>



Let Now Summarized Our Finding

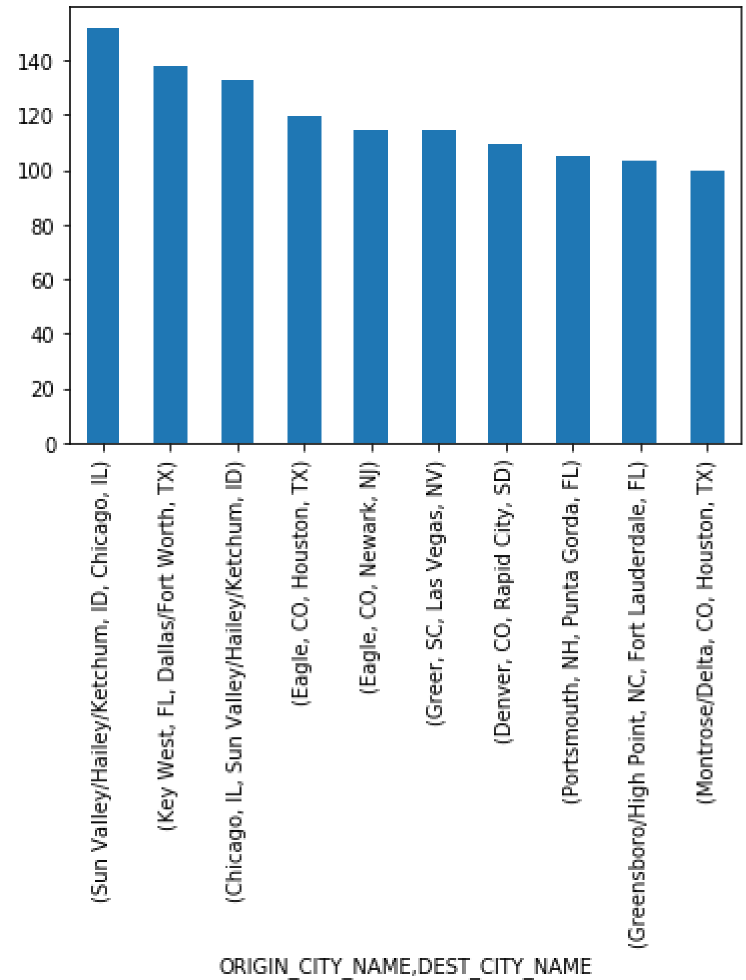
In [103]:
```
trip_with_delay_2017.plot(kind='bar', x='ORIGIN_CITY_NAME', y='ARR_DELA
Y')
```

Out[103]: <matplotlib.axes._subplots.AxesSubplot at 0x152870650>



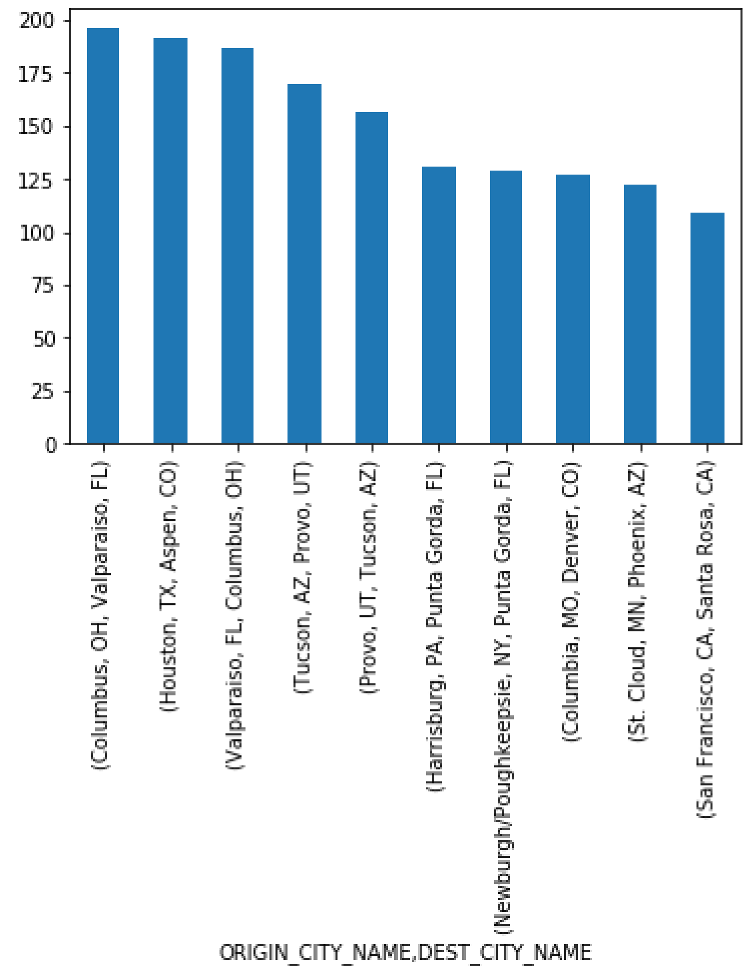
In [104]:
```
trip_with_delay_2018.plot(kind='bar', x='ORIGIN_CITY_NAME', y='ARR_DELAY')
```

Out[104]: <matplotlib.axes._subplots.AxesSubplot at 0x17a124990>



In [105]:
```
trip_with_delay_2019.plot(kind='bar', x='ORIGIN_CITY_NAME', y='ARR_DELA
Y')
```

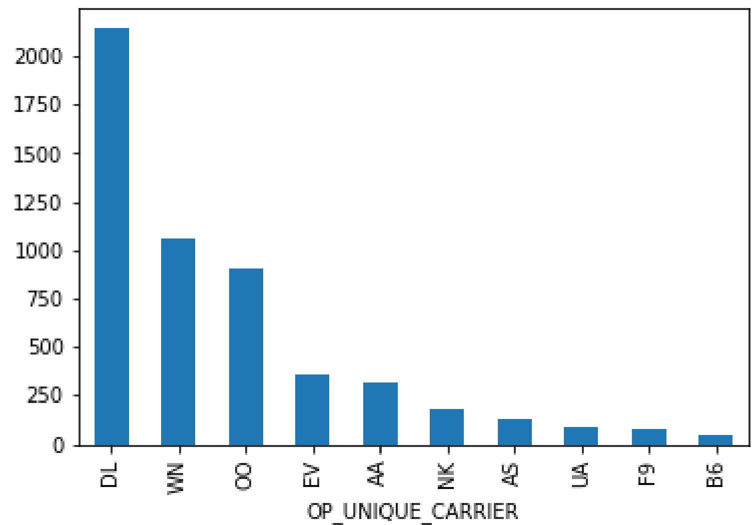
Out[105]: <matplotlib.axes._subplots.AxesSubplot at 0x17a237a50>



Our finding shows the trips that has the most delays in 2017 is different than 2018 and 2019

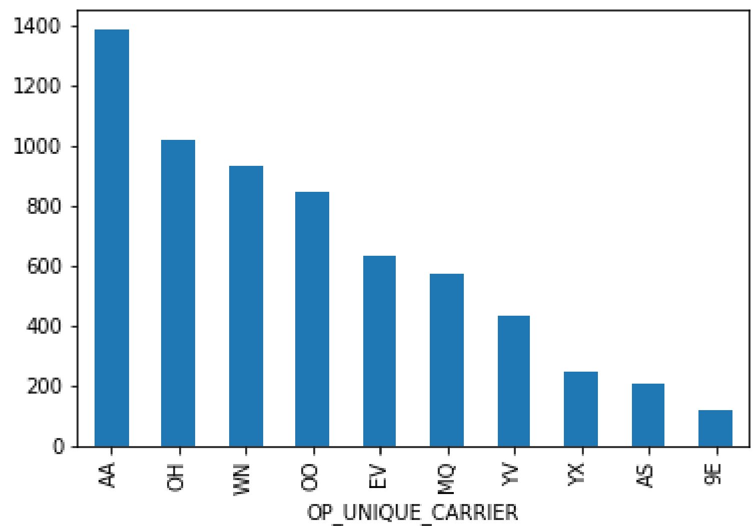
In [107]:
```
airline_with_most_cancellations_2017.plot(kind='bar', x='OP_UNIQUE_CARRIER')
```

Out[107]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x17a44a8d0>
```



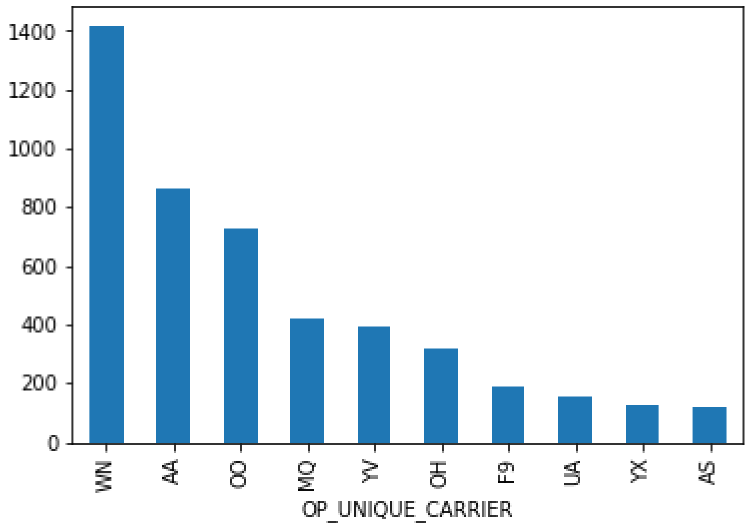
In [108]:
```
airline_with_most_cancellations_2018.plot(kind='bar', x='OP_UNIQUE_CARRIER')
```

Out[108]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x180699350>
```



In [109]:
```
airline_with_most_cancellations_2019.plot(kind='bar', x='OP_UNIQUE_CARRI
ER')
```

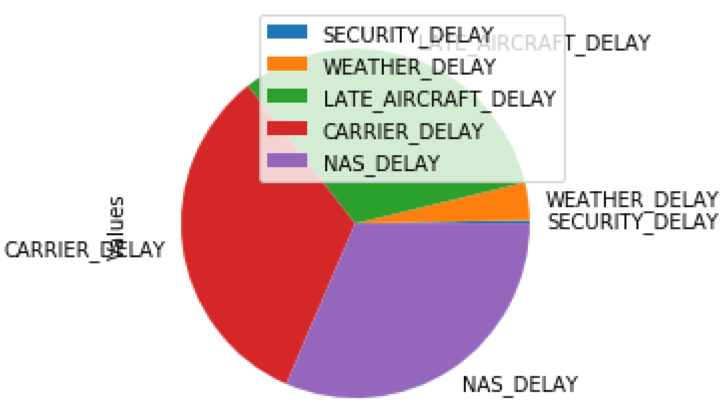
Out[109]: <matplotlib.axes._subplots.AxesSubplot at 0x1806909d0>



Here We can see some airline comming back, such as AA(American Airline),WN(Southwest Airlines),AS(Alaska Airlines) that have the most cancellation those past 3 years

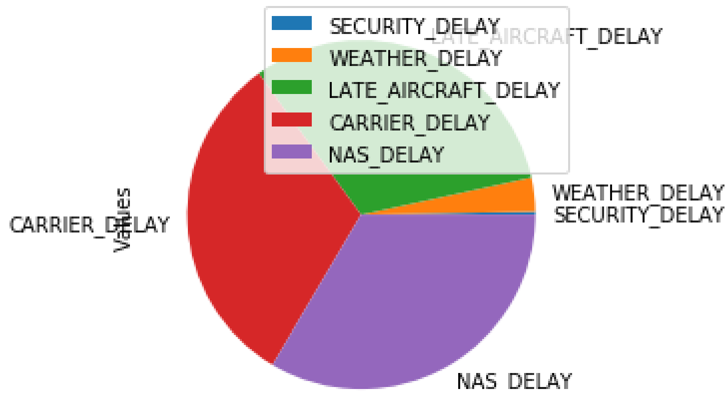
In [110]:
```
df_delays_causes_2017.plot(kind='pie',x="Delays Causes", y="Values")
```

Out[110]: <matplotlib.axes._subplots.AxesSubplot at 0x180708390>



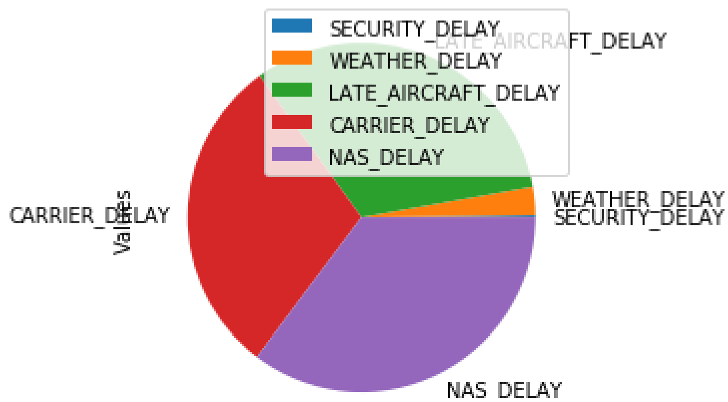
In [111]:
```
df_delays_causes_2018.plot(kind='pie',x="Delays Causes", y="Values")
```

Out[111]: <matplotlib.axes._subplots.AxesSubplot at 0x11d4f9e10>



In [112]:
```
df_delays_causes_2019.plot(kind='pie',x="Delays Causes", y="Values")
```

Out[112]: <matplotlib.axes._subplots.AxesSubplot at 0x1524813d0>



The Data clearly shows that the reason of the delays is due to the Carrier, Weather, National Air System

Our Prediction

We will use linear regression to predict base on the data we have if delays will decrease with time or not

first let merge all the data together

In [113]:
```
all_data = data_2017.append(data_2018).append(data_2019)
```

In [114]:
```
all_data.head()
```

Out[114]:

	YEAR	QUARTER	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	FL_DATE	OP_UNIQUE_CARRIER
0	2017	4	12	1	5	2017-12-01	OO
1	2017	4	12	1	5	2017-12-01	OO
2	2017	4	12	1	5	2017-12-01	OO
3	2017	4	12	1	5	2017-12-01	OO
4	2017	4	12	1	5	2017-12-01	OO

5 rows × 40 columns

In [116]:
```
import statsmodels.api as sm
```

We will Replace All NaN with 0 (Data Cleaning)

In [131]:
```
all_data_clean = all_data.fillna(0)
```

In [133]:
```
all_data.head()
```

Out[133]:

	YEAR	QUARTER	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	FL_DATE	OP_UNIQUE_CARRIER
0	2017	4	12	1	5	2017-12-01	OO
1	2017	4	12	1	5	2017-12-01	OO
2	2017	4	12	1	5	2017-12-01	OO
3	2017	4	12	1	5	2017-12-01	OO
4	2017	4	12	1	5	2017-12-01	OO

5 rows × 40 columns

In [158]:
```
prediction_aircraft_delay = sm.OLS(all_data["LATE_AIRCRAFT_DELAY"],all_d
ata["WEATHER_DELAY"]).fit()
```

In [159]:
```
prediction_security_delay.summary()
```

Out[159]:

OLS Regression Results

Dep. Variable:	LATE_AIRCRAFT_DELAY	**R-squared (uncentered):**	0.001
Model:	OLS	**Adj. R-squared (uncentered):**	0.001
Method:	Least Squares	**F-statistic:**	1457.
Date:	Thu, 19 Dec 2019	**Prob (F-statistic):**	1.06e-318
Time:	04:39:42	**Log-Likelihood:**	-7.6692e+06
No. Observations:	1694061	**AIC:**	1.534e+07
Df Residuals:	1694060	**BIC:**	1.534e+07
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
WEATHER_DELAY	0.0503	0.001	38.174	0.000	0.048	0.053

Omnibus:	3115034.989	**Durbin-Watson:**	1.778
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	12453197505.752
Skew:	13.376	**Prob(JB):**	0.00
Kurtosis:	422.178	**Cond. No.**	1.00

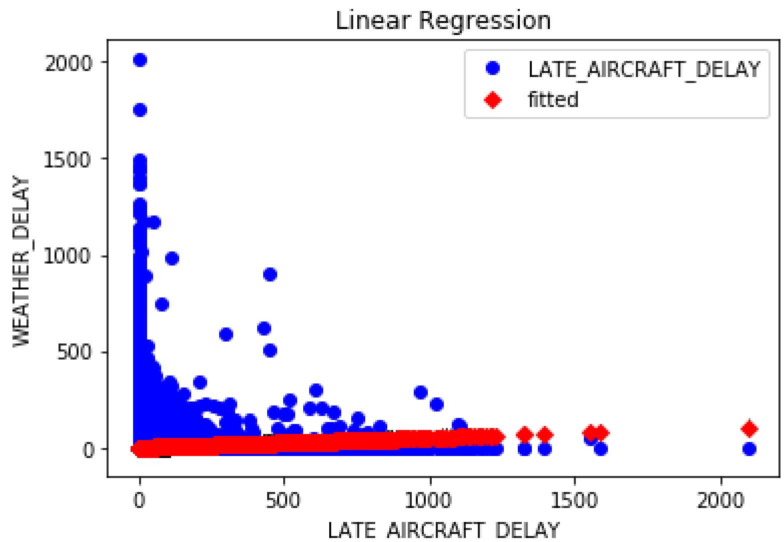
Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

using the code in the [documentation (https://www.statsmodels.org/stable/generated/statsmodels.graphi](https://www.statsmodels.org/stable/generated/statsmodels.graphi) we can vizualize our linear regression

In [160]:
```
import matplotlib.pyplot as plt
```

In [161]:
```
fig, ax = plt.subplots()
fig = sm.graphics.plot_fit(prediction_security_delay, 0, ax=ax)
ax.set_ylabel("WEATHER_DELAY")
ax.set_xlabel("LATE_AIRCRAFT_DELAY")
ax.set_title("Linear Regression")
```

Out[161]:
```
Text(0.5, 1.0, 'Linear Regression')
```



In [162]:
```
plt.show()
```

here in this relation between weather_delay and late_aircraft we can see clearly that the fitted point going up. So our prediction shows that the delay won't get better

In [163]:
```
prediction_carrier_delay = sm.OLS(all_data["CARRIER_DELAY"],all_data["WE
ATHER_DELAY"]).fit()
```

In [164]:
```
prediction_carrier_delay.summary()
```

Out[164]:

OLS Regression Results

Dep. Variable:	CARRIER_DELAY	**R-squared (uncentered):**	0.000
Model:	OLS	**Adj. R-squared (uncentered):**	0.000
Method:	Least Squares	**F-statistic:**	21.60
Date:	Thu, 19 Dec 2019	**Prob (F-statistic):**	3.37e-06
Time:	04:40:32	**Log-Likelihood:**	-8.0395e+06
No. Observations:	1694061	**AIC:**	1.608e+07
Df Residuals:	1694060	**BIC:**	1.608e+07
Df Model:	1		
Covariance Type:	nonrobust		

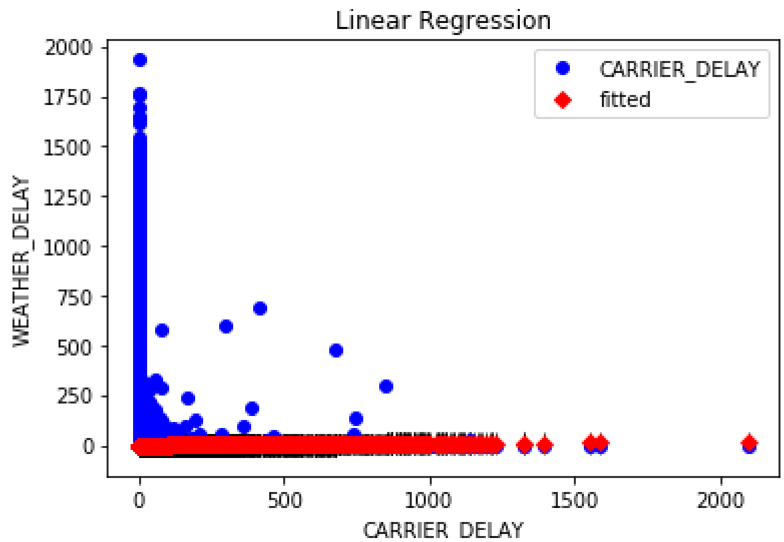
	coef	std err	t	P>|t|	[0.025	0.975]
WEATHER_DELAY	0.0076	0.002	4.647	0.000	0.004	0.011

Omnibus:	3910825.712	**Durbin-Watson:**	1.903
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	40821705195.753
Skew:	22.664	**Prob(JB):**	0.00
Kurtosis:	762.126	**Cond. No.**	1.00

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [167]:
```
fig, ax = plt.subplots()
fig = sm.graphics.plot_fit(prediction_carrier_delay, 0, ax=ax)
ax.set_ylabel("WEATHER_DELAY")
ax.set_xlabel("CARRIER_DELAY")
ax.set_title("Linear Regression")
```

Out[167]: `Text(0.5, 1.0, 'Linear Regression')`



In [168]:
```
plt.show()
```

our data predict that the CARRIER_DELAY will remain the same

In [ ]: