In [1]:
```python
import numpy as np
import pandas as pd
import seaborn as sns
import pandas_datareader.data as web
import datetime
```

In [2]:
```python
stockdata = pd.read_csv('csv/prices.csv')
```

In [3]:
```python
stockdata
```

Out[3]:

	date	symbol	open	close	low	high	volume
0	2016-01-05 00:00:00	WLTW	123.430000	125.839996	122.309998	126.250000	2163600.0
1	2016-01-06 00:00:00	WLTW	125.239998	119.980003	119.940002	125.540001	2386400.0
2	2016-01-07 00:00:00	WLTW	116.379997	114.949997	114.930000	119.739998	2489500.0
3	2016-01-08 00:00:00	WLTW	115.480003	116.620003	113.500000	117.440002	2006300.0
4	2016-01-11 00:00:00	WLTW	117.010002	114.970001	114.089996	117.330002	1408600.0
...	...	...	...	...	...	...	...
851259	2016-12-30	ZBH	103.309998	103.199997	102.849998	103.930000	973800.0
851260	2016-12-30	ZION	43.070000	43.040001	42.689999	43.310001	1938100.0
851261	2016-12-30	ZTS	53.639999	53.529999	53.270000	53.740002	1701200.0
851262	2016-12-30 00:00:00	AIV	44.730000	45.450001	44.410000	45.590000	1380900.0
851263	2016-12-30 00:00:00	FTV	54.200001	53.630001	53.389999	54.480000	705100.0

851264 rows × 7 columns

In [4]:
```python
    stockdata.sort_values(by='date')
```

Out[4]:

	date	symbol	open	close	low	high	volume
646	2010-01-04	SYMC	18.040001	18.400000	18.010000	18.530001	8322300.0
472	2010-01-04	IDXX	53.700001	54.080002	53.430000	54.349998	325200.0
473	2010-01-04	IFF	41.509998	42.009998	41.500000	42.020000	286000.0
474	2010-01-04	ILMN	31.120001	30.549999	30.420000	31.230000	1793700.0
475	2010-01-04	INTC	20.790001	20.879999	20.730000	21.030001	47800900.0
...	...	...	...	...	...	...	...
850932	2016-12-30	EXPE	113.839996	113.279999	112.720001	114.239998	2082700.0
851260	2016-12-30	ZION	43.070000	43.040001	42.689999	43.310001	1938100.0
250	2016-12-30 00:00:00	WLTW	122.589996	122.279999	121.389999	123.559998	466400.0
851262	2016-12-30 00:00:00	AIV	44.730000	45.450001	44.410000	45.590000	1380900.0
851263	2016-12-30 00:00:00	FTV	54.200001	53.630001	53.389999	54.480000	705100.0

851264 rows × 7 columns

In [5]:
```python
desc=stockdata.describe()
```

In [6]:
```python
desc
```

Out[6]:

	open	close	low	high	volume
count	851264.000000	851264.000000	851264.000000	851264.000000	8.512640e+05
mean	70.836986	70.857109	70.118414	71.543476	5.415113e+06
std	83.695876	83.689686	82.877294	84.465504	1.249468e+07
min	0.850000	0.860000	0.830000	0.880000	0.000000e+00
25%	33.840000	33.849998	33.480000	34.189999	1.221500e+06
50%	52.770000	52.799999	52.230000	53.310001	2.476250e+06
75%	79.879997	79.889999	79.110001	80.610001	5.222500e+06
max	1584.439941	1578.130005	1549.939941	1600.930054	8.596434e+08

In [ ]:

In [22]:
```python
google = stockdata[stockdata['symbol']=='GOOG']
google = google.set_index('date')
```

In [23]:
```python
google
```

Out[23]:

date	symbol	open	close	low	high	volume
2010-01-04	GOOG	626.951088	626.751061	624.241073	629.511067	3927000.0
2010-01-05	GOOG	627.181073	623.991055	621.541045	627.841071	6031900.0
2010-01-06	GOOG	625.861078	608.261023	606.361042	625.861078	7987100.0
2010-01-07	GOOG	609.401025	594.101005	592.651008	610.001045	12876600.0
2010-01-08	GOOG	592.000997	602.021036	589.110988	603.251034	9483900.0
...	...	...	...	...	...	...
2016-12-23	GOOG	790.900024	789.909973	787.280029	792.739990	623400.0
2016-12-27	GOOG	790.679993	791.549988	787.656982	797.859985	789100.0
2016-12-28	GOOG	793.700012	785.049988	783.200012	794.229980	1132700.0
2016-12-29	GOOG	783.330017	782.789978	778.919983	785.929993	742200.0
2016-12-30	GOOG	782.750000	771.820007	770.409973	782.780029	1760200.0

1762 rows × 6 columns

In [9]:
```python
tesla = stockdata[stockdata['symbol']=='TSLA']
```

In [10]:
```python
tesla.set_index('date')
```

Out[10]:

date	symbol	open	close	low	high	volume

In [11]:
```python
microsoft = stockdata[stockdata['symbol']=='MSFT']
microsoft = microsoft.set_index('date')
```

In [12]:
```python
microsoft
```

Out[12]:

date	symbol	open	close	low	high	volume
2010-01-04	MSFT	30.620001	30.950001	30.590000	31.100000	38409100.0
2010-01-05	MSFT	30.850000	30.959999	30.639999	31.100000	49749600.0
2010-01-06	MSFT	30.879999	30.770000	30.520000	31.080000	58182400.0
2010-01-07	MSFT	30.629999	30.450001	30.190001	30.700001	50559700.0
2010-01-08	MSFT	30.280001	30.660000	30.240000	30.879999	51197400.0
...	...	...	...	...	...	...
2016-12-23	MSFT	63.450001	63.240002	62.799999	63.540001	12403800.0
2016-12-27	MSFT	63.209999	63.279999	63.209999	64.070000	11763200.0
2016-12-28	MSFT	63.400002	62.990002	62.830002	63.400002	14653300.0
2016-12-29	MSFT	62.860001	62.900002	62.730000	63.200001	10250600.0
2016-12-30	MSFT	62.959999	62.139999	62.029999	62.990002	25579900.0

1762 rows × 6 columns

In [13]:
```python
ge = stockdata[stockdata['symbol']=='GE'].set_index('date')
```

In [14]:
```python
ge
```

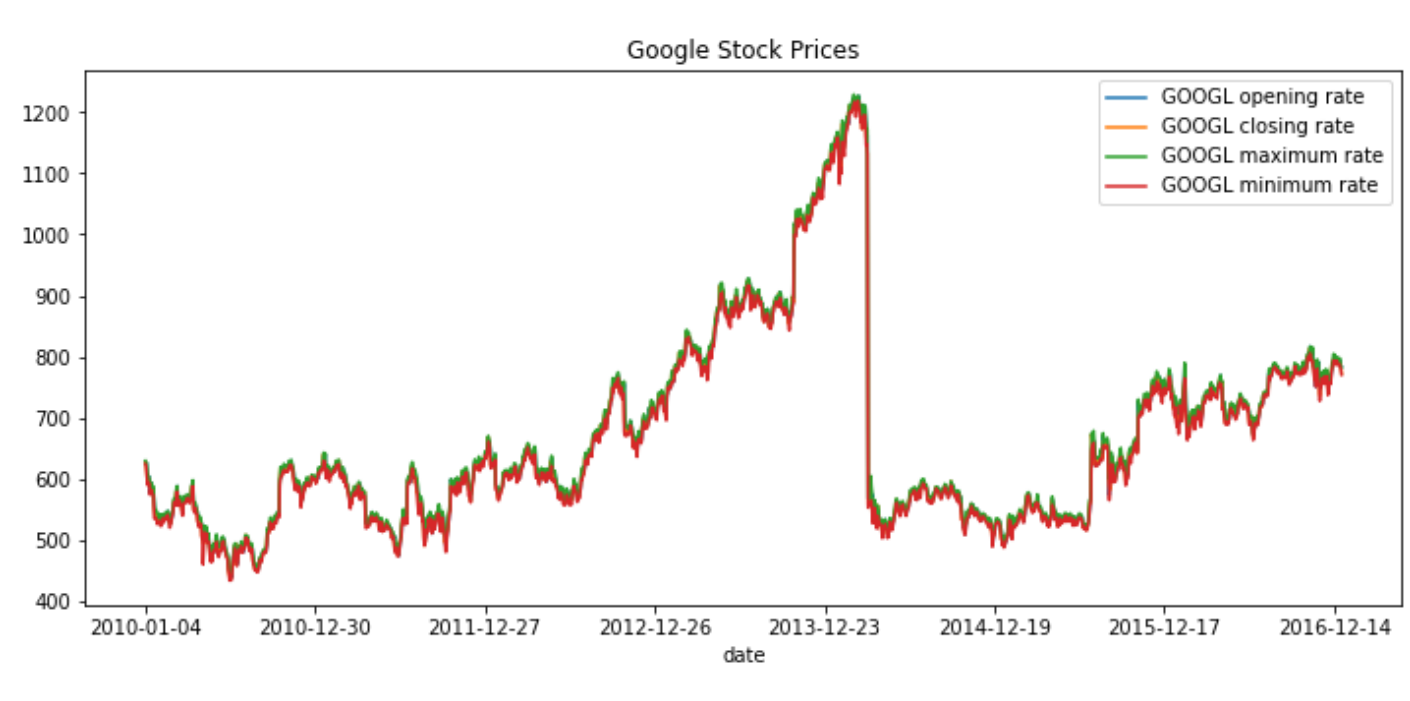
Out[14]:

date	symbol	open	close	low	high	volume
2010-01-04	GE	15.220000	15.450000	15.150000	15.640000	67079900.0
2010-01-05	GE	15.460000	15.530000	15.450000	15.670000	64550600.0
2010-01-06	GE	15.530000	15.450000	15.440000	15.620000	55464900.0
2010-01-07	GE	15.480000	16.250000	15.430000	16.480000	185472300.0
2010-01-08	GE	16.309999	16.600000	16.270000	16.690001	115112600.0
...	...	...	...	...	...	...
2016-12-23	GE	31.870001	31.879999	31.770000	31.940001	14559700.0
2016-12-27	GE	31.889999	31.900000	31.850000	32.049999	15630400.0
2016-12-28	GE	31.840000	31.700001	31.670000	31.969999	18444400.0
2016-12-29	GE	31.740000	31.709999	31.700001	31.879999	16072300.0
2016-12-30	GE	31.629999	31.600000	31.510000	31.799999	25350900.0

1762 rows × 6 columns

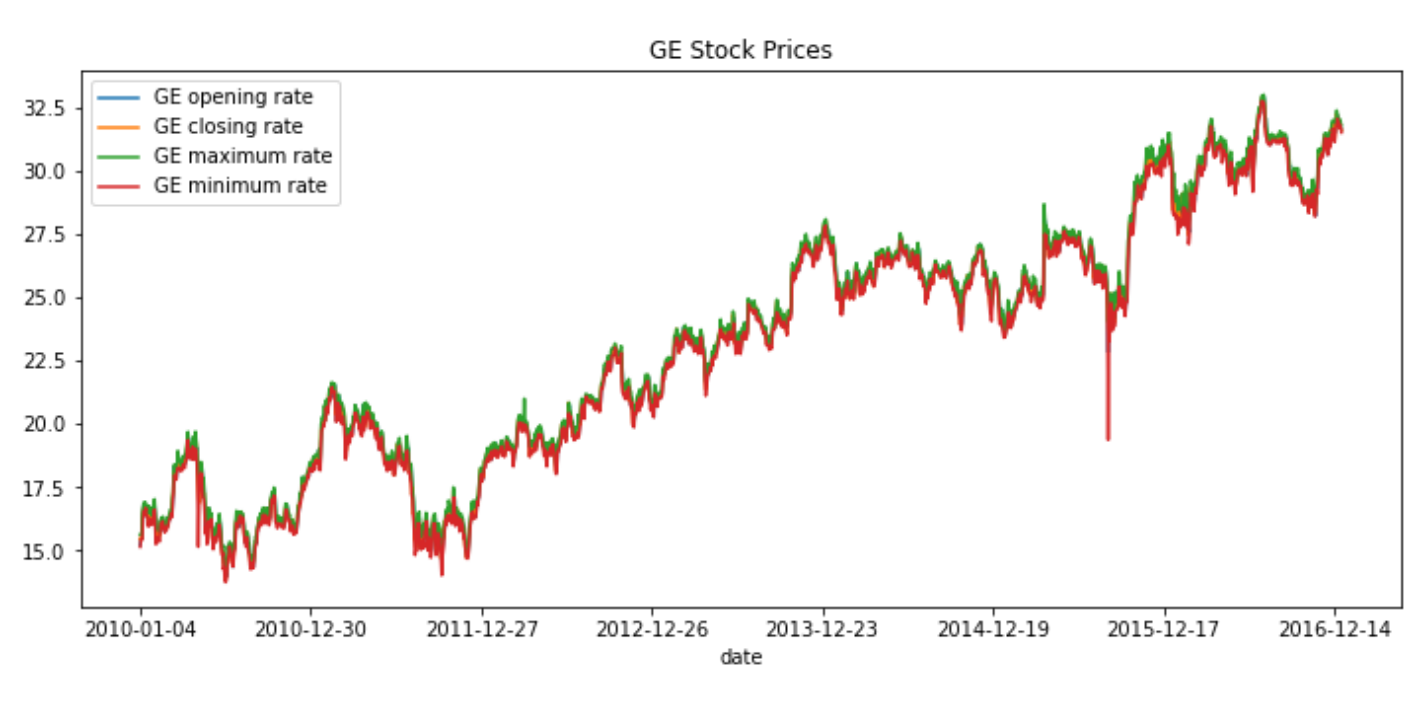
In [24]:
```python
google['open'].plot(label = 'GOOGL opening rate',legend = True, figsize = (12,5),ti
tle='Google Stock Prices')
google['close'].plot(label = 'GOOGL closing rate',legend = True, figsize = (12,5))
google['high'].plot(label = 'GOOGL maximum rate',legend = True, figsize = (12,5))
google['low'].plot(label = 'GOOGL minimum rate',legend = True, figsize = (12,5))
```

Out[24]: `<AxesSubplot:title={'center':'Google Stock Prices'}, xlabel='date'>`



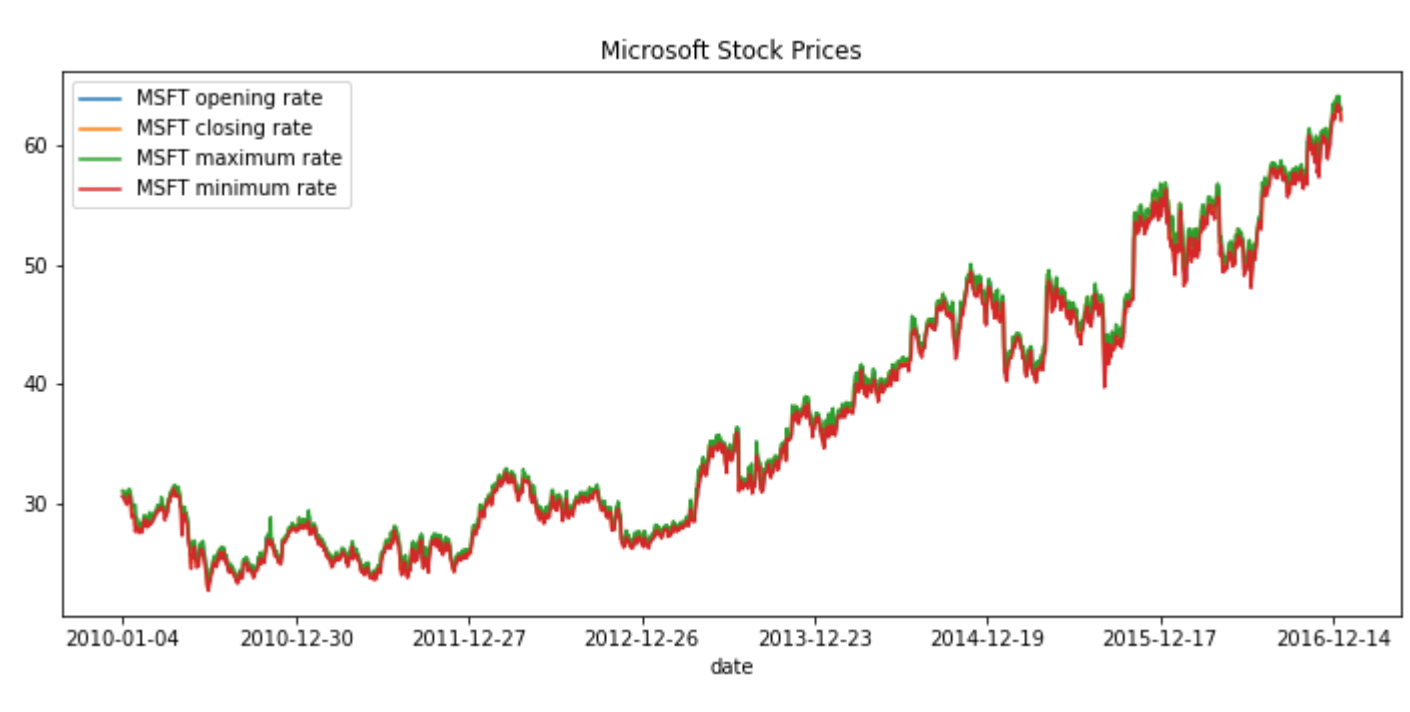
In [16]:
```python
ge['open'].plot(label = 'GE opening rate',legend = True, figsize = (12,5),title='GE
Stock Prices')
ge['close'].plot(label = 'GE closing rate',legend = True, figsize = (12,5))
ge['high'].plot(label = 'GE maximum rate',legend = True, figsize = (12,5))
ge['low'].plot(label = 'GE minimum rate',legend = True, figsize = (12,5))
```

Out[16]: `<AxesSubplot:title={'center':'GE Stock Prices'}, xlabel='date'>`



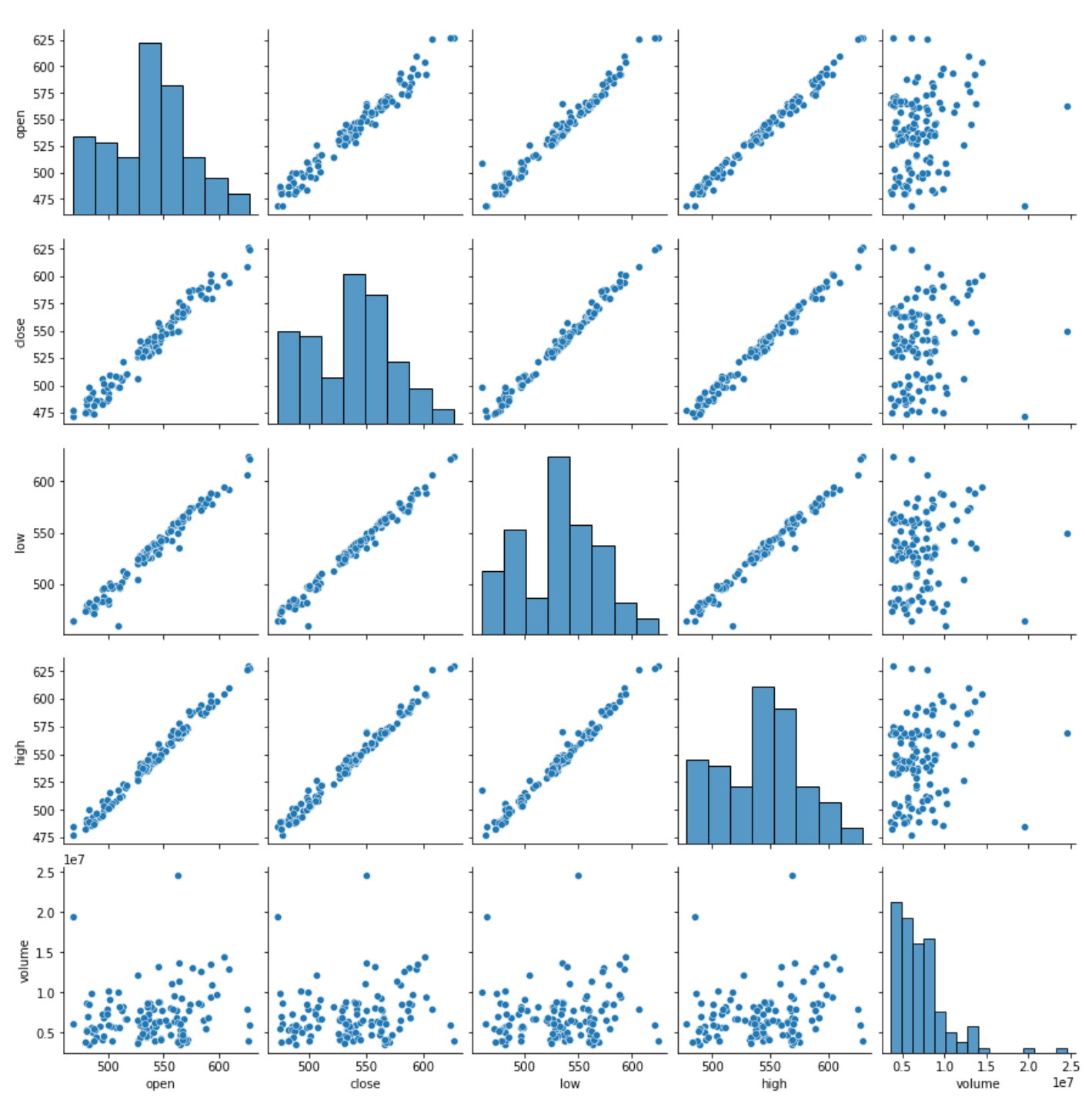
In [17]:
```python
microsoft['open'].plot(label = 'MSFT opening rate',legend = True, figsize = (12,5),
title='Microsoft Stock Prices')
microsoft['close'].plot(label = 'MSFT closing rate',legend = True, figsize = (12,5
))
microsoft['high'].plot(label = 'MSFT maximum rate',legend = True, figsize = (12,5))
microsoft['low'].plot(label = 'MSFT minimum rate',legend = True, figsize = (12,5))
```

Out[17]: `<AxesSubplot:title={'center':'Microsoft Stock Prices'}, xlabel='date'>`



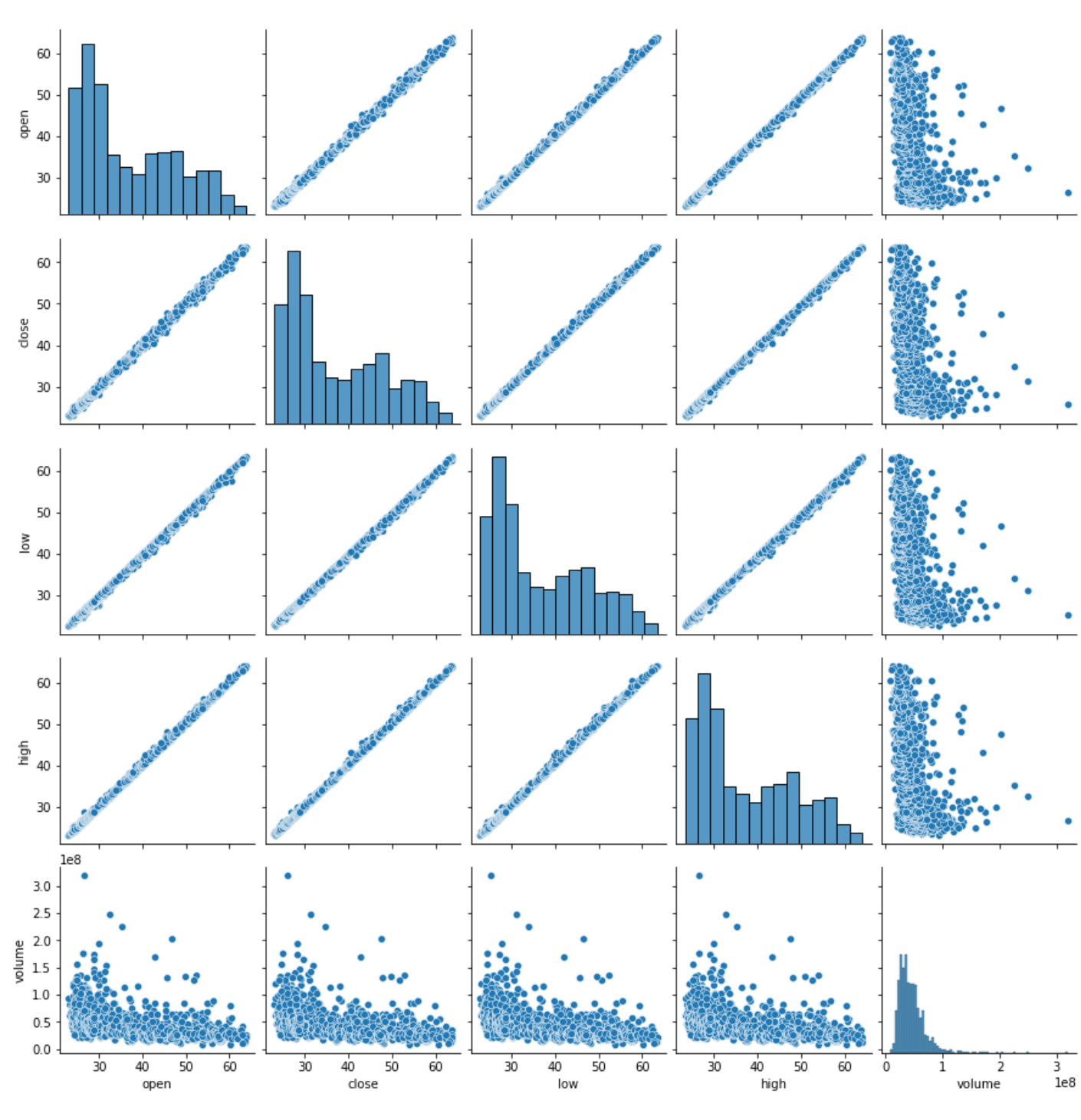
In [18]:
```python
sns.pairplot(google)
```

Out[18]: `<seaborn.axisgrid.PairGrid at 0x7fcfac1492e8>`



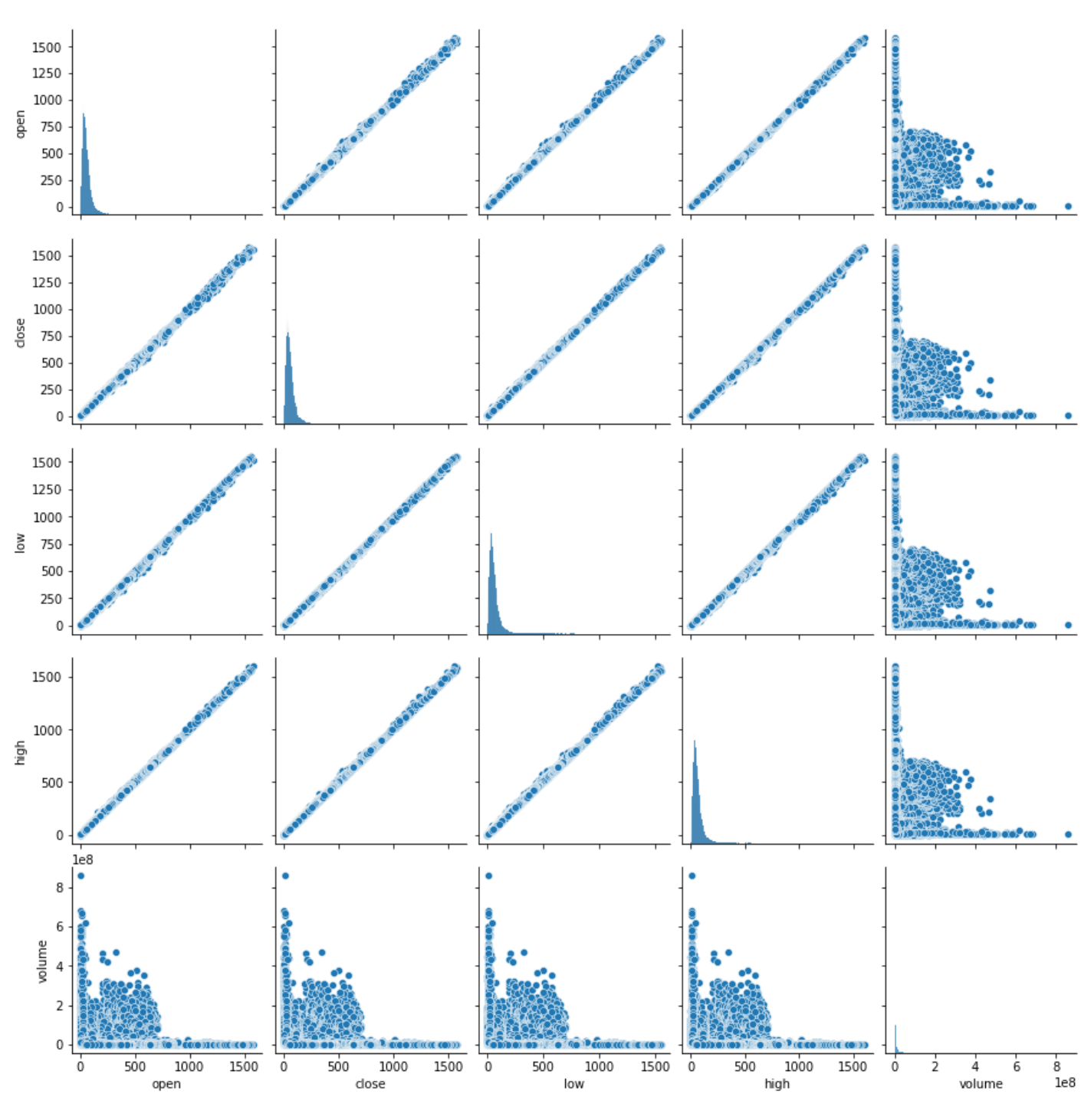
In [19]:
```python
sns.pairplot(microsoft)
```

Out[19]: `<seaborn.axisgrid.PairGrid at 0x7fcfa4f81080>`



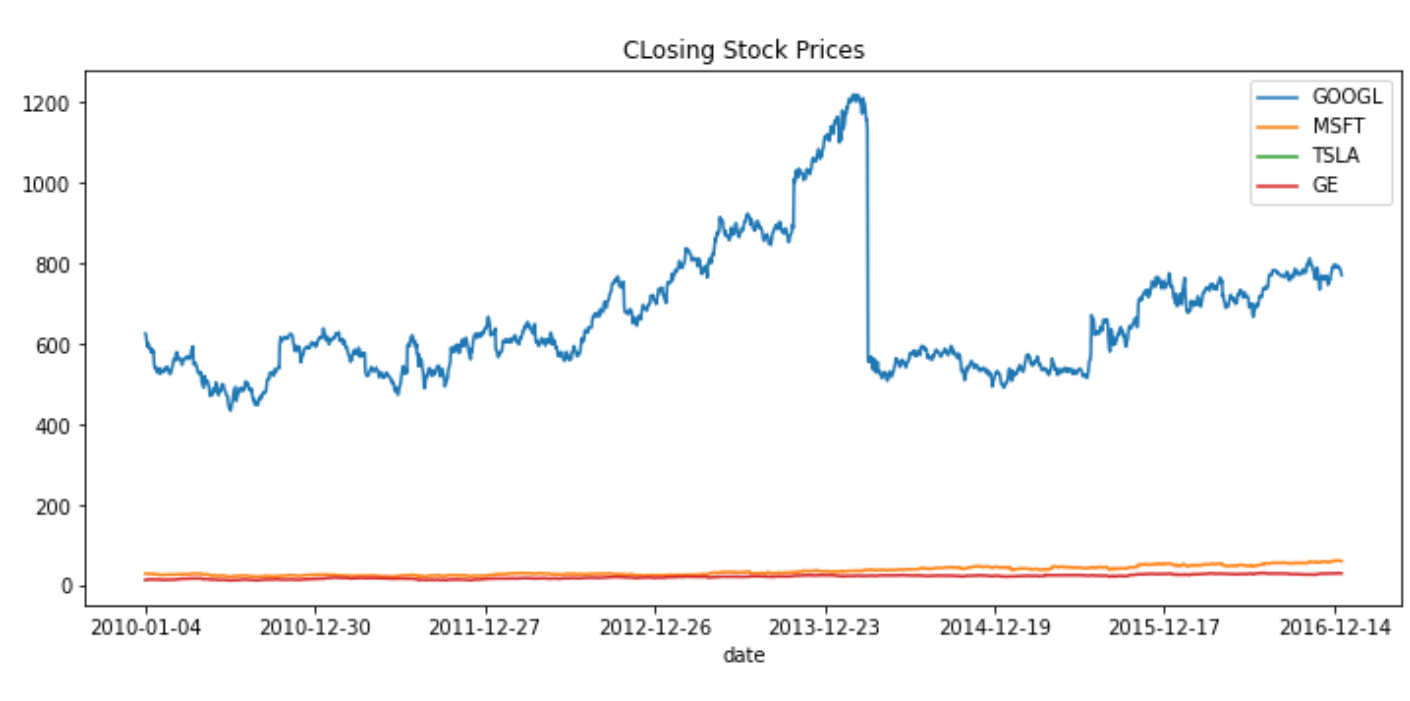
In [20]:
```python
sns.pairplot(stockdata)
```

Out[20]: `<seaborn.axisgrid.PairGrid at 0x7fcfa429fef0>`



In [57]:
```python
google['close'].plot(label = 'GOOGL',legend = True, figsize = (12,5),title='CLosing
Stock Prices')
microsoft['close'].plot(label = 'MSFT',legend = True, figsize = (12,5))
tesla['close'].plot(label = 'TSLA',legend = True, figsize = (12,5))
ge['close'].plot(label = 'GE',legend = True, figsize = (12,5))
```

Out[57]: `<AxesSubplot:title={'center':'CLosing Stock Prices'}, xlabel='date'>`



In [52]:
```python
newdf = pd.DataFrame(columns=["GOOG","MSFT","GE"])
newdf['GOOG'] =google['close']
newdf['MSFT'] = microsoft['close']
newdf['GE'] = ge['close']
```

In [53]:
```python
newdf
```

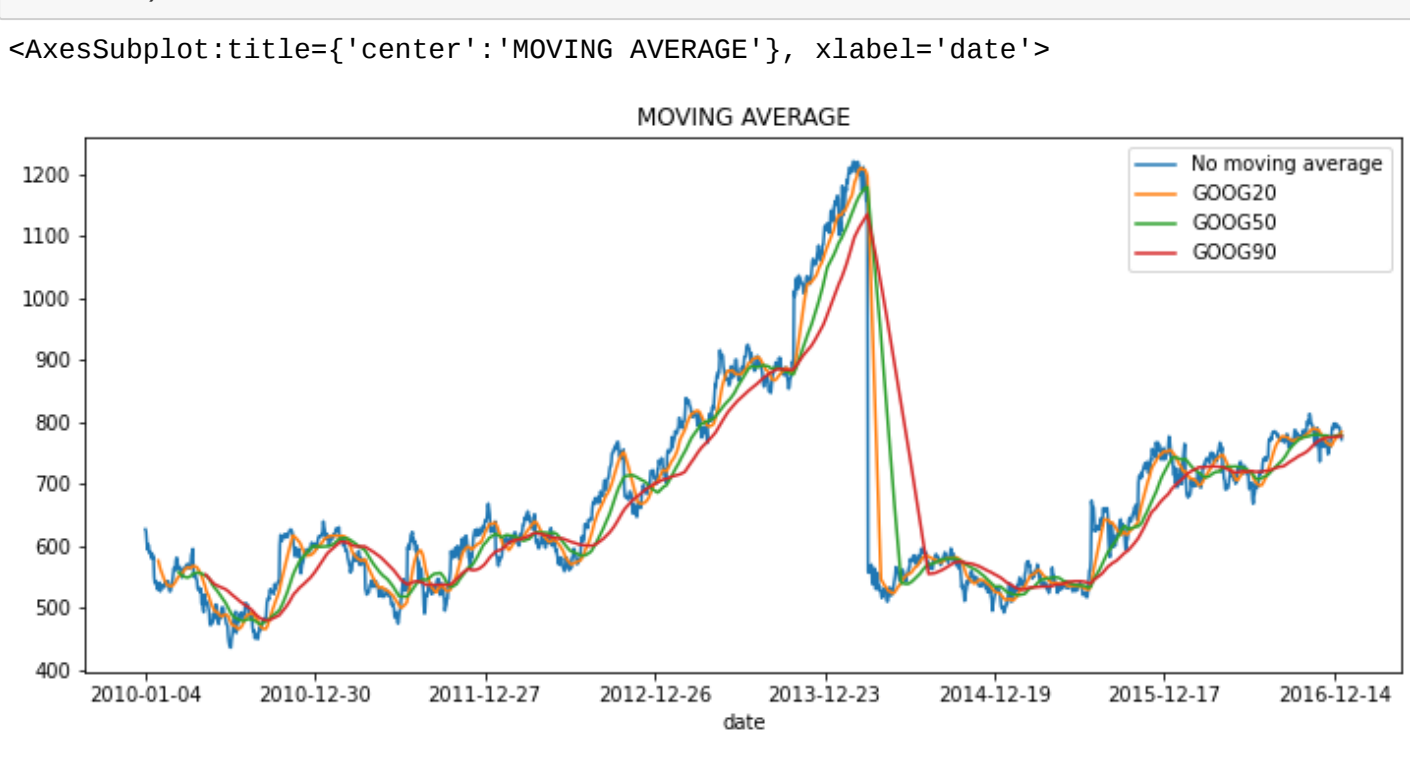
Out[53]:

date	GOOG	MSFT	GE
2010-01-04	626.751061	30.950001	15.450000
2010-01-05	623.991055	30.959999	15.530000
2010-01-06	608.261023	30.770000	15.450000
2010-01-07	594.101005	30.450001	16.250000
2010-01-08	602.021036	30.660000	16.600000
...	...	...	...
2016-12-23	789.909973	63.240002	31.879999
2016-12-27	791.549988	63.279999	31.900000
2016-12-28	785.049988	62.990002	31.700001
2016-12-29	782.789978	62.900002	31.709999
2016-12-30	771.820007	62.139999	31.600000

1762 rows × 3 columns

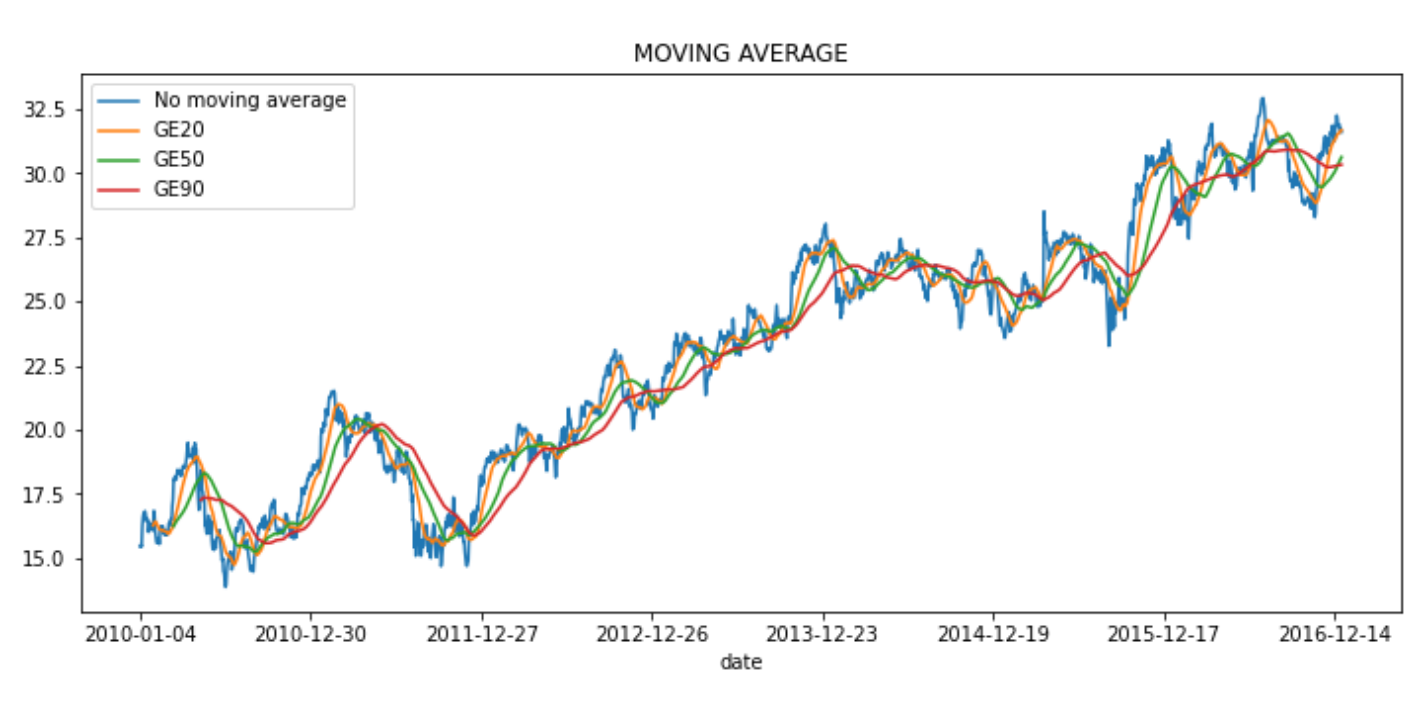
In [62]:
```python
#Calculating Simple moving average
newdf['GOOG'].plot(label = 'No moving average',legend = True, figsize = (12,5),titl
e='CLosing Stock Prices')
newdf['GOOG20'] = newdf['GOOG'].rolling(20).mean()
newdf['GOOG20'].plot(label = 'GOOG20',legend = True, figsize = (12,5),title='CLosin
g Stock Prices')
newdf['GOOG50'] = newdf['GOOG'].rolling(50).mean()
newdf['GOOG50'].plot(label = 'GOOG50',legend = True, figsize = (12,5),title='CLosin
g Stock Prices')
newdf['GOOG90'] = newdf['GOOG'].rolling(90).mean()
newdf['GOOG90'].plot(label = 'GOOG90',legend = True, figsize = (12,5),title='MOVING
AVERAGE')
```

Out[62]: `<AxesSubplot:title={'center':'MOVING AVERAGE'}, xlabel='date'>`



In [63]:
```python
#Calculating Simple moving average
newdf['GE'].plot(label = 'No moving average',legend = True, figsize = (12,5),title=
'CLosing Stock Prices')
newdf['GE20'] = newdf['GE'].rolling(20).mean()
newdf['GE20'].plot(label = 'GE20',legend = True, figsize = (12,5),title='CLosing St
ock Prices')
newdf['GE50'] = newdf['GE'].rolling(50).mean()
newdf['GE50'].plot(label = 'GE50',legend = True, figsize = (12,5),title='CLosing St
ock Prices')
newdf['GE90'] = newdf['GE'].rolling(90).mean()
newdf['GE90'].plot(label = 'GE90',legend = True, figsize = (12,5),title='MOVING AVE
RAGE')
```

Out[63]: `<AxesSubplot:title={'center':'MOVING AVERAGE'}, xlabel='date'>`



In [65]:
```python
import sys
if "\your\path\to\xelatex" not in sys.path:
    print('adding path') # I just add this to know if the path was present or not.
    sys.path.append("\your\path\to\xelatex")
```

```
  File "<ipython-input-65-3711ee5406d8>", line 2
    if "\your\path\to\xelatex" not in sys.path:
      ^
SyntaxError: (unicode error) 'unicodeescape' codec can't decode bytes in position 1
3-15: truncated \xXX escape
```

In [ ]: