Extracting the city level data and Exporting to CSV:

select * from city_data where city='Cairo' and country='Egypt';

Extracting the global data. Exporting to CSV:

select * from global_data

Prepare the data to be visualized in the chart using python:

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

In [2]:
```
paths1="C:\\Users\\El Zahraa\\Desktop\\code\\my folder\\Data Analysis_udacity\\results (1).csv"
paths2="C:\\Users\\El Zahraa\\Desktop\\code\\my folder\\Data Analysis_udacity\\results (2).csv"
```

In [3]:
```
df1 = pd.read_csv(paths1)
df1.rename(columns={'avg_temp':'avg_temp_local'},inplace=True)
df2 = pd.read_csv(paths2)
df2.rename(columns={'avg_temp':'avg_temp_global'},inplace=True)
```

Calculating the moving average for local temperature:

In [4]:
```
df1['MA_temp_local']=df1.iloc[:,3].rolling(window=10).mean()
df1.head()
```

Out[4]:

	year	city	country	avg_temp_local	MA_temp_local
0	1808	Cairo	Egypt	17.11	NaN
1	1809	Cairo	Egypt	19.87	NaN
2	1810	Cairo	Egypt	19.93	NaN
3	1811	Cairo	Egypt	20.00	NaN
4	1812	Cairo	Egypt	19.93	NaN

In [5]:
```
df2.head()
```

Out[5]:

	year	avg_temp_global
0	1750	8.72
1	1751	7.98
2	1752	5.78
3	1753	8.39
4	1754	8.47

Calculating the moving average for global temperature:

In [6]:
```
x=df2['year'].to_list()
x.index(1808)
df2_modified = df2[['avg_temp_global']][x.index(1808):x.index(2014)]
df2_modified.reset_index(drop=True,inplace=True)
df2_modified['MA_temp_global']=df2_modified.iloc[:,0].rolling(window=10).mean()
df2_modified.tail()
```

Out[6]:

	avg_temp_global	MA_temp_global
201	9.51	9.493
202	9.70	9.543
203	9.52	9.554
204	9.51	9.548
205	9.81	9.556

In [7]:
```
df = pd.concat([df1,df2_modified],axis=1)
```

In [8]:
```
df.head(20)
```

Out[8]:

	year	city	country	avg_temp_local	MA_temp_local	avg_temp_global	MA_temp_global
0	1808	Cairo	Egypt	17.11	NaN	7.63	NaN
1	1809	Cairo	Egypt	19.87	NaN	7.08	NaN
2	1810	Cairo	Egypt	19.93	NaN	6.92	NaN
3	1811	Cairo	Egypt	20.00	NaN	6.86	NaN
4	1812	Cairo	Egypt	19.93	NaN	7.05	NaN
5	1813	Cairo	Egypt	20.51	NaN	7.74	NaN
6	1814	Cairo	Egypt	20.43	NaN	7.59	NaN
7	1815	Cairo	Egypt	20.30	NaN	7.24	NaN
8	1816	Cairo	Egypt	20.51	NaN	6.94	NaN
9	1817	Cairo	Egypt	21.88	20.047	6.98	7.203
10	1818	Cairo	Egypt	11.60	19.496	7.83	7.223
11	1819	Cairo	Egypt	20.31	19.540	7.37	7.252
12	1820	Cairo	Egypt	20.58	19.605	7.62	7.322
13	1821	Cairo	Egypt	20.63	19.668	8.09	7.445
14	1822	Cairo	Egypt	20.72	19.747	8.19	7.559
15	1823	Cairo	Egypt	20.71	19.767	7.72	7.557
16	1824	Cairo	Egypt	21.44	19.868	8.55	7.653
17	1825	Cairo	Egypt	21.00	19.938	8.39	7.768
18	1826	Cairo	Egypt	20.94	19.981	8.36	7.910
19	1827	Cairo	Egypt	21.63	19.956	8.81	8.093

In [9]:
```
df.tail(10)
```

Out[9]:

	year	city	country	avg_temp_local	MA_temp_local	avg_temp_global	MA_temp_global
196	2004	Cairo	Egypt	22.08	22.001	9.32	9.343
197	2005	Cairo	Egypt	22.01	22.046	9.70	9.378
198	2006	Cairo	Egypt	22.05	22.063	9.53	9.427
199	2007	Cairo	Egypt	22.36	22.154	9.73	9.480
200	2008	Cairo	Egypt	22.64	22.182	9.43	9.471
201	2009	Cairo	Egypt	22.63	22.217	9.51	9.493
202	2010	Cairo	Egypt	23.72	22.440	9.70	9.543
203	2011	Cairo	Egypt	21.99	22.406	9.52	9.554
204	2012	Cairo	Egypt	22.48	22.413	9.51	9.548
205	2013	Cairo	Egypt	22.91	22.487	9.61	9.556

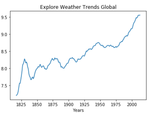
Data Visualization:

In [10]:
```
plt.plot(df['year'],df['MA_temp_local'])
plt.xlabel('Years')
plt.ylabel('Moving Average Temperature (Local)')
plt.title('Explore Weather Trends Local(Cairo)')
```

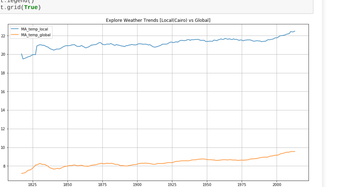
Out[10]: Text(0.5, 1.0, 'Explore Weather Trends Local(Cairo)')

In [11]:
```
plt.plot(df['year'],df['MA_temp_global'])
plt.xlabel('Years')
plt.ylabel('Moving Average Temperature (Global)')
plt.title('Explore Weather Trends Global')
```

Out[11]: Text(0.5, 1.0, 'Explore Weather Trends Global')



In [12]:
```
plt.figure(figsize=(15,8))
plt.plot(df['year'],df['MA_temp_local'],label='MA_temp_local')
plt.plot(df['year'],df['MA_temp_global'],label='MA_temp_global')
plt.xlabel('Years')
plt.ylabel('Moving Average Temperature')
plt.title('Explore Weather Trends (Local(Cairo) vs Global)')
plt.legend()
plt.grid(True)
```



Observations:

- As we see from the chart that Egypt(Cairo) average temperature is hotter than global average temperature.
- Difference between local and average temperature is almost constant.
- For local average, the change in temperature is almost constant over 200 years between 20 and 22.
- For global average, the change in temperature for the 19th century is almost constant, but for the 20th century it is observed that temperature is getting higher.
- As we see in the previous point that the world is getting hotter which it is called "Global Warming" in the 21th cenyury.
- As well, we see in local average that temperature is getting hotter starting from 2000 (21th century) and that is a correct indicator about the change in temperature for the average of the global temperature.
- As we will see in the next section that calculating correlation between data will confirm the previous two points

Save modified data to new csv file:

In [13]:
```
df.to_csv('total_temp.csv')
```

Correlation between data:

In [14]:
```
df[['year','MA_temp_local','MA_temp_global']].corr()
```

Out[14]:

	year	MA_temp_local	MA_temp_global
year	1.000000	0.866663	0.924778
MA_temp_local	0.866663	1.000000	0.932544
MA_temp_global	0.924778	0.932544	1.000000

In [15]:
```
from scipy import stats
df.dropna(subset=['MA_temp_local'],axis=0,inplace=True)
df.dropna(subset=['MA_temp_global'],axis=0,inplace=True)
```

In [16]:
```
pearson_coef, p_value = stats.pearsonr(df['MA_temp_local'], df['MA_temp_global'])
print(f"The Pearson Correlation Coefficient is {pearson_coef}")
print(f"with a P-value of P = {p_value}")
```

```
The Pearson Correlation Coefficient is 0.9325437121930521
with a P-value of P = 3.2633517931175094e-88
```

In [17]:
```
pearson_coef, p_value = stats.pearsonr(df['MA_temp_local'], df['year'])
print(f"The Pearson Correlation Coefficient is {pearson_coef}")
print(f"with a P-value of P = {p_value}")
```

```
The Pearson Correlation Coefficient is 0.8666630171361
with a P-value of P = 8.49965931579098e-61
```

In [18]:
```
pearson_coef, p_value = stats.pearsonr(df['MA_temp_global'], df['year'])
print(f"The Pearson Correlation Coefficient is {pearson_coef}")
print(f"with a P-value of P = {p_value}")
```

```
The Pearson Correlation Coefficient is 0.9247781432590557
with a P-value of P = 9.133072732434628e-84
```

Estimation the average temperature in your city based on the average global temperature:

In [19]:
```
from sklearn.linear_model import LinearRegression
```

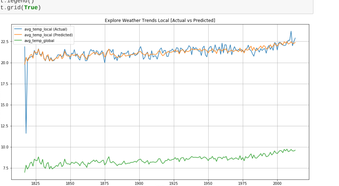
In [20]:
```
df.dtypes
```

Out[20]:

```
year                int64
city               object
country            object
avg_temp_local    float64
MA_temp_local     float64
avg_temp_global   float64
MA_temp_global    float64
dtype: object
```

In [21]:
```
lm=LinearRegression()
x=df[['avg_temp_global']]
y=df['avg_temp_local']
lm.fit(x,y)
yhat=lm.predict(x)
```

In [22]:
```
plt.figure(figsize=(15,8))
plt.plot(df['year'],y,label='avg_temp_local (Actual)')
plt.plot(df['year'],yhat,label='avg_temp_local (Predicted)')
plt.plot(df['year'],x,label='avg_temp_global')
plt.xlabel('Years')
plt.ylabel('Average Local Temperature')
plt.title("Explore Weather Trends Local (Actual vs Predicted)")
plt.legend()
plt.grid(True)
```



In [23]:
```
jupyter nbconvert thenotebook.ipynb --to latex
```

```
  File "<ipython-input-23-cad59af01fcc>", line 1
    jupyter nbconvert thenotebook.ipynb --to latex
          ^
SyntaxError: invalid syntax
```

In [ ]:
```

```