

UDACITY- Data Analyst Nanodegree**Project Submission: Explore Weather Trends**Tools:

SQL

Ms-Excel

Step 1: Accessing Data With SQLExtract data from a database using SQL

select * from global_data

The screenshot shows a web-based SQL interface. At the top, there's a header with 'Input', 'HISTORY', and 'MENU'. Below this, a schema explorer on the left lists 'city_data', 'city_list', and 'global_data'. The main query editor contains the SQL statement 'select * from global_data'. A green 'Success!' message and an 'EVALUATE' button are visible. The 'Output' section shows '266 results' and a 'Download CSV' link. Below this is a table with two columns: 'year' and 'avg_temp'. The table displays data for years 1750 through 1757. At the bottom, there are 'Menu' and 'Expand' buttons.

year	avg_temp
1750	8.72
1751	7.98
1752	5.78
1753	8.39
1754	8.47
1755	8.36
1756	8.85
1757	9.02

Observation: We got 266 results after running SQL query.

select * from city_list

02.06.2020

where country='India'

The screenshot shows a web-based SQL interface. The 'Input' section contains a schema explorer on the left with a tree view showing 'city_data', 'city_list', 'city', and 'country' under 'city_data', and 'global_data' at the root. The main query editor contains the SQL: `select * from city_list where country='India'`. A 'Success!' message and an 'EVALUATE' button are visible. The 'Output' section shows '22 results' and a 'Download CSV' link. Below this is a table with two columns: 'city' and 'country'. The table lists 22 cities, all of which are in India. The cities listed are: Agra, Ahmadabad, Allahabad, Amritsar, Bangalore, Bhopal, Delhi, and Haora. The interface also has a dark footer bar with 'Menu' and 'Expand' buttons.

city	country
Agra	India
Ahmadabad	India
Allahabad	India
Amritsar	India
Bangalore	India
Bhopal	India
Delhi	India
Haora	India

Observation: We got 22 results after running SQL query. Data of 22 cities of India provided in global_data database.

Our nearest big city is "Jaipur". Now we will Alter columns 'avg_temp'. This column is contained by both city_data and global_data.

```
ALTER TABLE city_data RENAME COLUMN avg_temp to city_avg_temp;
```

```
ALTER TABLE global_data RENAME COLUMN avg_temp to global_avg_temp;
```

02.06.2020

Input

HISTORY ▾


MENU ▾

SCHEMA		
city_data	▾	ALTER TABLE city_data RENAME COLUMN avg_temp to city_avg_temp;
city_list	▴	ALTER TABLE global_data RENAME COLUMN avg_temp to global_avg_temp;
city		
country		
global_data	▾	Success!

EVALUATE

Output

No data to download


ALTER Operation Successful

▴ Menu

⌵ Expand

Join the two tables and have the relevant data

```
SELECT global_data.year, global_data.global_avg_temp, city_data.city_avg_temp
FROM global_data
JOIN city_data
ON global_data.year = city_data.year
WHERE city LIKE 'Jaipur';
```

02.06.2020

Input

HISTORY ▾

MENU ▾

SCHEMA ↻

city_data ▾

city_list ▲

city

country

global_data ▾

```
SELECT global_data.year, global_data.global_avg_temp,
city_data.city_avg_temp
FROM global_data
JOIN city_data
ON global_data.year = city_data.year
WHERE city LIKE 'Jaipur';
```

Success!

EVALUATE

Output 218 results

Download CSV

year	global_avg_temp	city_avg_temp
1796	8.27	25.25
1797	8.51	26.77
1798	8.67	24.46
1799	8.51	25.51
1800	8.48	25.44
1801	8.59	24.38
1802	8.58	25.87
1803	8.50	25.62

^ Menu ↗ Expand

Download the file as CSV Format. Downloaded as “results-4.csv”.

Here, we got 218 results starting from year 1796 for both.

Step2: Moving averages

Calculated Moving averages to use in the line chart

Note: We delete the missing year data

14	1808	7.63	
15	1809	7.08	
16	1810	6.92	
17	1811	6.86	
18	1812	7.05	

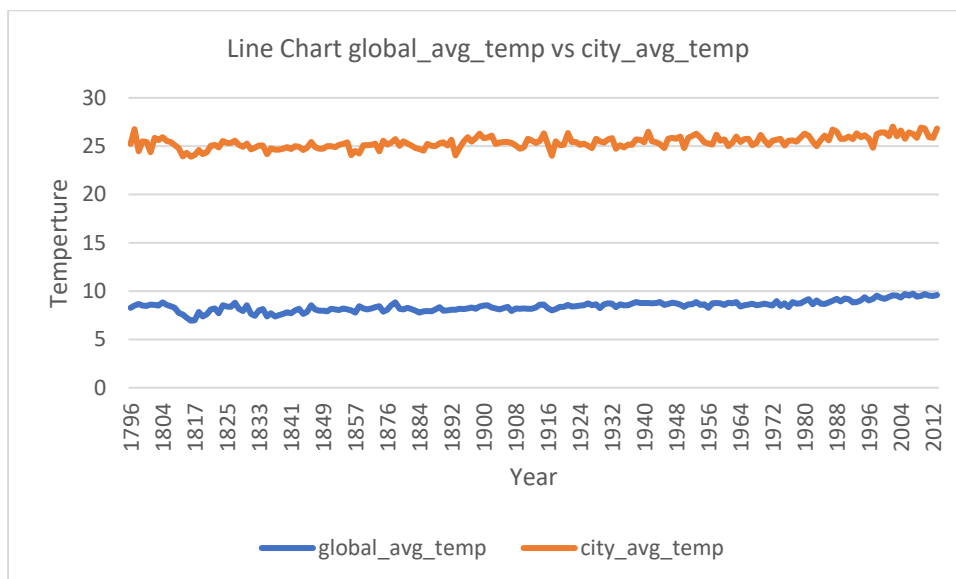
02.06.2020

Now We calculate Moving Average for the first 10 years

AVERAGE		X		✓	f _x	=AVERAGE(B2:C11)
	A	B	C	D	E	F
1	year	global_avg_temp	city_avg_temp	Moving Average		
2	1796	8.27	25.25			
3	1797	8.51	26.77			
4	1798	8.67	24.46			
5	1799	8.51	25.51			
6	1800	8.48	25.44			
7	1801	8.59	24.38			
8	1802	8.58	25.87			
9	1803	8.5	25.62			
10	1804	8.84	25.93			
11	1805	8.56	25.54	=AVERAGE(B2:C11)		
12	1806	8.43	25.45	AVERAGE(number1, [number2], ...)		
13	1807	8.28	25.09			

AVERAGE (B2:C11) = 17.014

Plot Line Chart global_avg_temp vs city_avg_temp



Observations:

1. Chart shows a difference between Jaipur city's average temperature and global average temperature. Jaipur city have temperatures greater than the global average.
2. Jaipur city seems to be hotter than any other cities in world.
3. Since India lies near to equator and all other cities lies between equator so have high temperatures as compare to global average.
4. From chart global temperature rise is constant.