databricks DE Casestudy

(https://databricks.com)

```
from datetime import date
from dateutil.relativedelta import relativedelta

days_in_year=365
date_list=[]

for day in range(days_in_year):
    date_list.append(date.today() + relativedelta(days=-day))


import requests

cities=['Mumbai','Delhi','Kolkata','London','Chennai','Bangalore','Kochi','Gandhinagar','Amaravati','Itanagar','Shimla']


%python
raw_data_list=[]
city_data={}
for day in date_list:
    for city in cities:
        url=f"http://api.weatherapi.com/v1/history.json?key=f[redacted]5&q={city}&dt={day}"
        response = requests.get(url)

        city_info={}
        raw_data=response.json()
        for key, value in dict.items(raw_data['location']):
            if key not in ['localtime_epoch','localtime']:
                   #print(f"{key} : {value}")
                   city_info[key]=value

        city_data[raw_data['location']['name']]=city_info
        raw_data_list.append(raw_data)



  ConnectTimeout: HTTPConnectionPool(host='api.weatherapi.com', port=80): Max retries exceeded with url: /v1/history.jso
  n?key=f274a0add27347cc8fd161013232605&q=Mumbai&dt=2023-05-20 (Caused by ConnectTimeoutError(, 'Connection to api.weathe
  rapi.com timed out. (connect timeout=None)'))

raw_data_df=spark.read.json(sc.parallelize(raw_data_list))
display(raw_data_df)
```

Table

	forecast	location
	▸ {"forecastday": [{"astro": {"moon_illumination": "44", "moon_phase": "First Quarter", "moonrise": "12:22 PM", "moonset": "12:47 AM", "sunrise": "06:01 AM", "sunset": "07:11 PM"}, "date": "2023-05-27", "date_epoch": 1685145600, "day": {"avghumidity": 67, "avgtemp_c": 30, "avgtemp_f": 85.9, "avgvis_km": 10, "avgvis_miles": 6, "condition": {"code": 1063, "icon": "//cdn.weatherapi.com/weather/64x64/day/176.png", "text": "Patchy rain possible"}, "maxtemp_c": 32.5, "maxtemp_f": 90.5, "maxwind_kph": 20.9, "maxwind_mph": 13, "mintemp_c": 28.5, "mintemp_f": 83.3, "totalprecip_in": 0.01, "totalprecip_mm": 0.2, "uv": 7}, "hour": [{"chance_of_rain": 0, "chance_of_snow": 0, "cloud": 44, "condition": {"code": 1003, "icon": "//cdn.weatherapi.com/weather/64x64/night/116.png", "text": "Partly cloudy"}, "dewpoint_c": 22.6, "dewpoint_f": 72.7, "feelslike_c": 32.5, "feelslike_f": 90.5, "gust_kph": 23.8, "gust_mph": 14.8, "heatindex_c": 32.5, "heatindex_f": 90.5, "humidity": 68, "is_day": 0, "precip_in": 0, "precip_mm": 0, "pressure_in": 29.85, "pressure_mb": 1011, "temp_c": 29, "temp_f": 84.2, "time": "2023-05-27 00:00", "time_epoch": 1685125800, "uv": 1, "vis_km": 10, "vis_miles": 6, "will_it_rain": 0, "will_it_snow": 0, "wind_degree": 245, "wind_dir": "WSW", "wind_kph": 17.6, "wind_mph": 11, "windchill_c": 29, "windchill_f": 84.2}, {"chance_of_rain": 0, "chance_of_snow": 0, "cloud": 44, "condition": {"code": 1003, "icon": "//cdn.weatherapi.com/weather/64x64/night/116.png", "text": "Partly cloudy"}, "dewpoint_c": 22.4, "dewpoint_f": 72.3, "feelslike_c": 32.3, "feelslike_f": 90.1, "gust_kph": 22.3, "gust_mph": 13.9, "heatindex_c": 32.3, "heatindex_f": 90.1, "humidity": 68, "is_day": 0, "precip_in": 0, "precip_mm": 0, "pressure_in": 29.83, "pressure_mb": 1010, "temp_c": 28.9, "temp_f": 84, "time": "2023-05-27 01:00", "time_epoch": 1685129400, "uv": 1, "vis_km": 10, "vis_miles": 6, "will_it_rain": 0, "will_it_snow": 0, "wind_degree": 247, "wind_dir": "WSW", "wind_kph": 16.6, "wind_mph": 10.3, "windchill_c": 28.9, "windchill_f": 84}, {"chance_of_rain": 0, "chance_of_snow": 0, "cloud": 48, "condition": {"code": 1003, "icon": "//cdn.weatherapi.com/weather/64x64/night/116.png", "text": "Partly cloudy"}, "dewpoint_c": 22.3, "dewpoint_f": 72.1, "feelslike_c": 32.1, "feelslike_f": 89.8, "gust_kph": 19.8, "gust_mph": 12.3, "heatindex_c": 32.1, "heatindex_f": 89.8, "humidity": 68, "is_day": 0, "precip_in": 0, "precip_mm": 0, "pressure_in": 29.82, "pressure_mb": 1010, "temp_c": 28.8, "temp_f": 83.8, "time": "2023-05-27 02:00", "time_epoch": 1685133000, "uv": 1, "vis_km": 10, "vis_miles": 6, "will_it_rain": 0, "will_it_snow": 0, "wind_degree": 251,	▸ {"country": "India", "lat": "Mumbai", "region": "Mah

77 rows

```
raw_data_df.printSchema()
```

```
root
 |-- forecast: struct (nullable = true)
 |    |-- forecastday: array (nullable = true)
 |    |    |-- element: struct (containsNull = true)
 |    |    |    |-- astro: struct (nullable = true)
 |    |    |    |    |-- moon_illumination: string (nullable = true)
 |    |    |    |    |-- moon_phase: string (nullable = true)
 |    |    |    |    |-- moonrise: string (nullable = true)
 |    |    |    |    |-- moonset: string (nullable = true)
 |    |    |    |    |-- sunrise: string (nullable = true)
 |    |    |    |    |-- sunset: string (nullable = true)
 |    |    |    |-- date: string (nullable = true)
 |    |    |    |-- date_epoch: long (nullable = true)
 |    |    |    |-- day: struct (nullable = true)
 |    |    |    |    |-- avghumidity: double (nullable = true)
 |    |    |    |    |-- avgtemp_c: double (nullable = true)
 |    |    |    |    |-- avgtemp_f: double (nullable = true)
 |    |    |    |    |-- avgvis_km: double (nullable = true)
 |    |    |    |    |-- avgvis_miles: double (nullable = true)
 |    |    |    |    |-- condition: struct (nullable = true)
 |    |    |    |    |    |-- code: long (nullable = true)
```

```
display(raw_data_df)
```

Table

	forecast	location
	▶ {"forecastday": [{"astro": {"moon_illumination": "44", "moon_phase": "First Quarter", "moonrise": "12:22 PM", "moonset": "12:47 AM", "sunrise": "06:01 AM", "sunset": "07:11 PM"}, "date": "2023-05-27", "date_epoch": 1685145600, "day": {"avghumidity": 67, "avgtemp_c": 30, "avgtemp_f": 85.9, "avgvis_km": 10, "avgvis_miles": 6, "condition": {"code": 1063, "icon": "//cdn.weatherapi.com/weather/64x64/day/176.png", "text": "Patchy rain possible"}, "maxtemp_c": 32.5, "maxtemp_f": 90.5, "maxwind_kph": 20.9, "maxwind_mph": 13, "mintemp_c": 28.5, "mintemp_f": 83.3, "totalprecip_in": 0.01, "totalprecip_mm": 0.2, "uv": 7}, "hour": [{"chance_of_rain": 0, "chance_of_snow": 0, "cloud": 44, "condition": {"code": 1003, "icon": "//cdn.weatherapi.com/weather/64x64/night/116.png", "text": "Partly cloudy"}, "dewpoint_c": 22.6, "dewpoint_f": 72.7, "feelslike_c": 32.5, "feelslike_f": 90.5, "gust_kph": 23.8, "gust_mph": 14.8, "heatindex_c": 32.5, "heatindex_f": 90.5, "humidity": 68, "is_day": 0, "precip_in": 0, "precip_mm": 0, "pressure_in": 29.85, "pressure_mb": 1011, "temp_c": 29, "temp_f": 84.2, "time": "2023-05-27 00:00", "time_epoch": 1685125800, "uv": 1, "vis_km": 10, "vis_miles": 6, "will_it_rain": 0, "will_it_snow": 0, "wind_degree": 245, "wind_dir": "WSW", "wind_kph": 17.6, "wind_mph": 11, "windchill_c": 29, "windchill_f": 84.2}, {"chance_of_rain": 0, "chance_of_snow": 0, "cloud": 44, "condition": {"code": 1003, "icon": "//cdn.weatherapi.com/weather/64x64/night/116.png", "text": "Partly cloudy"}, "dewpoint_c": 22.4, "dewpoint_f": 72.3, "feelslike_c": 32.3, "feelslike_f": 90.1, "gust_kph": 22.3, "gust_mph": 13.9, "heatindex_c": 32.3, "heatindex_f": 90.1, "humidity": 68, "is_day": 0, "precip_in": 0, "precip_mm": 0, "pressure_in": 29.83, "pressure_mb": 1010, "temp_c": 28.9, "temp_f": 84, "time": "2023-05-27 01:00", "time_epoch": 1685129400, "uv": 1, "vis_km": 10, "vis_miles": 6, "will_it_rain": 0, "will_it_snow": 0, "wind_degree": 247, "wind_dir": "WSW", "wind_kph": 16.6, "wind_mph": 10.3, "windchill_c": 28.9, "windchill_f": 84}, {"chance_of_rain": 0, "chance_of_snow": 0, "cloud": 48, "condition": {"code": 1003, "icon": "//cdn.weatherapi.com/weather/64x64/night/116.png", "text": "Partly cloudy"}, "dewpoint_c": 22.3, "dewpoint_f": 72.1, "feelslike_c": 32.1, "feelslike_f": 89.8, "gust_kph": 19.8, "gust_mph": 12.3, "heatindex_c": 32.1, "heatindex_f": 89.8, "humidity": 68, "is_day": 0, "precip_in": 0, "precip_mm": 0, "pressure_in": 29.82, "pressure_mb": 1010, "temp_c": 28.8, "temp_f": 83.8, "time": "2023-05-27 02:00", "time_epoch": 1685133000, "uv": 1, "vis_km": 10, "vis_miles": 6, "will_it_rain": 0, "will_it_snow": 0, "wind_degree": 251,	▶ {"country": "India", "lat": "Mumbai", "region": "Mah

77 rows

```
df_avg=raw_data_df.rdd.map(lambda x: \
    (x.location["name"],x.location["country"],x.location["region"],x.forecast["forecastday"][0]
["date"],x.forecast["forecastday"][0]["day"])) \
    .toDF(["name","country","region","date","forecast"])
df_avg.printSchema()
df_avg.show()
```

```
root
 |-- name: string (nullable = true)
 |-- country: string (nullable = true)
 |-- region: string (nullable = true)
 |-- date: string (nullable = true)
 |-- forecast: struct (nullable = true)
 |    |-- avghumidity: double (nullable = true)
 |    |-- avgtemp_c: double (nullable = true)
 |    |-- avgtemp_f: double (nullable = true)
 |    |-- avgvis_km: double (nullable = true)
```

```
 |    |-- avgvis_miles: double (nullable = true)
 |    |-- condition: struct (nullable = true)
 |    |    |-- code: long (nullable = true)
 |    |    |-- icon: string (nullable = true)
 |    |    |-- text: string (nullable = true)
 |    |-- maxtemp_c: double (nullable = true)
 |    |-- maxtemp_f: double (nullable = true)
 |    |-- maxwind_kph: double (nullable = true)
 |    |-- maxwind_mph: double (nullable = true)
 |    |-- mintemp_c: double (nullable = true)
```

```
df_daily_data=df_avg.rdd.map(lambda x: \

(x.name,x.country,x.region,x.date,x.forecast.avghumidity,x.forecast.avgtemp_c,x.forecast.avgvis_km,x.forecast.condition["t
ext"],x.forecast.totalprecip_in,x.forecast.maxtemp_c,x.forecast.mintemp_c)) \

.toDF(["name","country","region","date","avghumidity","avgtemp_c","avgvis_km","condition","totalprecip_in","maxtemp_c","mi
ntemp_c"])
df_daily_data.printSchema()
df_daily_data.show()
```

```
root
 |-- name: string (nullable = true)
 |-- country: string (nullable = true)
 |-- region: string (nullable = true)
 |-- date: string (nullable = true)
 |-- avghumidity: double (nullable = true)
 |-- avgtemp_c: double (nullable = true)
 |-- avgvis_km: double (nullable = true)
 |-- condition: string (nullable = true)
 |-- totalprecip_in: double (nullable = true)
 |-- maxtemp_c: double (nullable = true)
 |-- mintemp_c: double (nullable = true)

+-----------+--------------+--------------------+----------+-----------+---------+---------+--------------------+--------
------+---------+----------+
|       name|       country|              region|      date|avghumidity|avgtemp_c|avgvis_km|           condition|totalpre
cip_in|maxtemp_c|mintemp_c|
+-----------+--------------+--------------------+----------+-----------+---------+---------+--------------------+--------
------+---------+----------+
|     Mumbai|         India|         Maharashtra|2023-05-27|       67.0|     30.0|     10.0|Patchy rain possible|
0.01|     32.5|     28.5|
```

```
df_daily_data.count()
```

```
Out[13]: 77
```

```
df2=raw_data_df.rdd.map(lambda x: \
    (x.location["name"],x.location["country"],x.location["region"],x.forecast["forecastday"][0]
["date"],x.forecast["forecastday"][0]["hour"])) \
    .toDF(["name","country","region","date","hourly"])
df2.printSchema()
df2.show()
```

```
root
 |-- name: string (nullable = true)
 |-- country: string (nullable = true)
 |-- region: string (nullable = true)
 |-- date: string (nullable = true)
 |-- hourly: array (nullable = true)
 |    |-- element: struct (containsNull = true)
 |    |    |-- chance_of_rain: long (nullable = true)
 |    |    |-- chance_of_snow: long (nullable = true)
 |    |    |-- cloud: long (nullable = true)
 |    |    |-- condition: struct (nullable = true)
 |    |    |    |-- code: long (nullable = true)
 |    |    |    |-- icon: string (nullable = true)
 |    |    |    |-- text: string (nullable = true)
 |    |    |-- dewpoint_c: double (nullable = true)
 |    |    |-- dewpoint_f: double (nullable = true)
 |    |    |-- feelslike_c: double (nullable = true)
```

```
 |    |    |-- feelslike_f: double (nullable = true)
 |    |    |-- gust_kph: double (nullable = true)
 |    |    |-- gust_mph: double (nullable = true)
 |    |    |-- heatindex_c: double (nullable = true)
```

```
from pyspark.sql.functions import explode
df_hourly=df2.select(df2.name,df2.country,df2.region,df2.date,explode(df2.hourly))
```

```
df_hourly.count()
df_hourly.printSchema()
```

```
root
 |-- name: string (nullable = true)
 |-- country: string (nullable = true)
 |-- region: string (nullable = true)
 |-- date: string (nullable = true)
 |-- col: struct (nullable = true)
 |    |-- chance_of_rain: long (nullable = true)
 |    |-- chance_of_snow: long (nullable = true)
 |    |-- cloud: long (nullable = true)
 |    |-- condition: struct (nullable = true)
 |    |    |-- code: long (nullable = true)
 |    |    |-- icon: string (nullable = true)
 |    |    |-- text: string (nullable = true)
 |    |-- dewpoint_c: double (nullable = true)
 |    |-- dewpoint_f: double (nullable = true)
 |    |-- feelslike_c: double (nullable = true)
 |    |-- feelslike_f: double (nullable = true)
 |    |-- gust_kph: double (nullable = true)
 |    |-- gust_mph: double (nullable = true)
 |    |-- heatindex_c: double (nullable = true)
 |    |-- heatindex_f: double (nullable = true)
```

```
df_hourly.show()
```

```
+------+-------+-----------+----------+--------------------+
|  name|country|     region|      date|                 col|
+------+-------+-----------+----------+--------------------+
|Mumbai|  India|Maharashtra|2023-05-27|{0, 0, 44, {1003,...|
|Mumbai|  India|Maharashtra|2023-05-27|{0, 0, 44, {1003,...|
|Mumbai|  India|Maharashtra|2023-05-27|{0, 0, 48, {1003,...|
|Mumbai|  India|Maharashtra|2023-05-27|{0, 0, 49, {1003,...|
|Mumbai|  India|Maharashtra|2023-05-27|{0, 0, 53, {1003,...|
|Mumbai|  India|Maharashtra|2023-05-27|{0, 0, 56, {1063,...|
|Mumbai|  India|Maharashtra|2023-05-27|{0, 0, 66, {1006,...|
|Mumbai|  India|Maharashtra|2023-05-27|{0, 0, 77, {1006,...|
|Mumbai|  India|Maharashtra|2023-05-27|{0, 0, 63, {1006,...|
|Mumbai|  India|Maharashtra|2023-05-27|{0, 0, 50, {1003,...|
|Mumbai|  India|Maharashtra|2023-05-27|{0, 0, 36, {1003,...|
|Mumbai|  India|Maharashtra|2023-05-27|{0, 0, 71, {1063,...|
|Mumbai|  India|Maharashtra|2023-05-27|{0, 0, 10, {1000,...|
|Mumbai|  India|Maharashtra|2023-05-27|{0, 0, 4, {1000, ...|
|Mumbai|  India|Maharashtra|2023-05-27|{0, 0, 1, {1000, ...|
|Mumbai|  India|Maharashtra|2023-05-27|{0, 0, 0, {1000, ...|
|Mumbai|  India|Maharashtra|2023-05-27|{0, 0, 0, {1000, ...|
|Mumbai|  India|Maharashtra|2023-05-27|{0, 0, 0, {1000, ...|
```

```
%sql
create database de_casestudy
```

```
  Error in SQL statement: NamespaceAlreadyExistsException: [SCHEMA_ALREADY_EXISTS] Cannot create schema `de_casestudy` be
  cause it already exists.
  Choose a different name, drop the existing schema, or add the IF NOT EXISTS clause to tolerate pre-existing schema.
```

```
df_daily_data.write.saveAsTable('de_casestudy.daily_avg_data')
```

```
  AnalysisException: [TABLE_OR_VIEW_ALREADY_EXISTS] Cannot create table or view `de_casestudy`.`daily_avg_data` because i
  t already exists.
  Choose a different name, drop or replace the existing object, add the IF NOT EXISTS clause to tolerate pre-existing obj
  ects, or add the OR REFRESH clause to refresh the existing streaming table.
```

```
%sql
use database de_casestudy
```

OK

```
%sql
show tables
```

Table

	database	tableName	isTemporary
1	de_casestudy	daily_avg_data	false

1 row

```
%sql
select * from daily_avg_data
```

Table

	name	country	region	date	avghumidity	avgtemp_c	avgvis_km	cond
1	Mumbai	India	Maharashtra	2023-05-21	70	29.8	10	Partly
2	Delhi	Canada	Ontario	2023-05-21	72	14.9	10	Sunn
3	Kolkata	India	West Bengal	2023-05-21	62	33.3	10	Partly
4	London	United Kingdom	City of London, Greater London	2023-05-21	73	13.5	10	Over
5	Chennai	India	Tamil Nadu	2023-05-21	74	31.7	9.8	Partly
6	Bangalore	India	Karnataka	2023-05-21	60	27.2	9.6	Patch
7	Kochi	India	Kerala	2023-05-21	77	28.8	9.3	Mod

133 rows

```
%sql
show tables
```

Table

	database	tableName	isTemporary
1	de_casestudy	daily_avg_data	false

1 row

OK

```
df_hourly.write.saveAsTable('de_casestudy.temp_hourly_data')
```

```
%sql
--name,country,region,date,col.chance_of_rain as chance_of_rain,col.chance_of_snow as chance_of_snow,
col.condition["text"] as condition_description
select name,country,region,date,col.chance_of_rain as chance_of_rain,col.chance_of_snow as chance_of_snow,
col.condition["text"] as condition_description , col.feelslike_c as feelslike_c, col.heatindex_c as heatindex_c,
col.temp_c as temp_c ,col.time as measurement_time from hourly_data
```

```
Error in SQL statement: AnalysisException: [TABLE_OR_VIEW_NOT_FOUND] The table or view `hourly_data` cannot be found. V
erify the spelling and correctness of the schema and catalog.
If you did not qualify the name with a schema, verify the current_schema() output, or qualify the name with the correct
schema and catalog.
To tolerate the error on drop use DROP VIEW IF EXISTS or DROP TABLE IF EXISTS.; line 2 pos 276;
'Project ['name, 'country, 'region, 'date, 'col.chance_of_rain AS chance_of_rain#4446, 'col.chance_of_snow AS chance_of
_snow#4447, 'col.condition[text] AS condition_description#4448, 'col.feelslike_c AS feelslike_c#4449, 'col.heatindex_c
AS heatindex_c#4450, 'col.temp_c AS temp_c#4451, 'col.time AS measurement_time#4452]
+- 'UnresolvedRelation [hourly_data], [], false
```

```
%sql
create table daily_hourly_data as
select name,country,region,date,col.chance_of_rain as chance_of_rain,col.chance_of_snow as chance_of_snow,
col.condition["text"] as condition_description , col.feelslike_c as feelslike_c, col.heatindex_c as heatindex_c,
col.temp_c as temp_c ,col.time as measurement_time from temp_hourly_data
```

Query returned no results

```
%sql
select * from daily_hourly_data
```

Table

	name	country	region	date	chance_of_rain	chance_of_snow	condition_des
1	Kolkata	India	West Bengal	2023-05-22	0	0	Clear
2	Kolkata	India	West Bengal	2023-05-22	0	0	Clear
3	Kolkata	India	West Bengal	2023-05-22	0	0	Clear
4	Kolkata	India	West Bengal	2023-05-22	0	0	Clear
5	Kolkata	India	West Bengal	2023-05-22	0	0	Clear
6	Kolkata	India	West Bengal	2023-05-22	0	0	Sunny
7	Kolkata	India	West Bengal	2023-05-22	0	0	Sunny

1,848 rows

```
daily_avg_weather_data=spark.read.table('daily_avg_data')
```

```
daily_hourly_data=spark.read.table('daily_hourly_data')
```

```
daily_avg_weather_data.write.option("header",True)\
   .csv("/tmp/spark_output/daily_avg_weather_data")
```

```
daily_hourly_data.write.option("header",True)\
   .csv("/tmp/spark_output/daily_hourly_data")
```

```
spark.read.csv("/tmp/spark_output/daily_avg_weather_data",header=True).show()
```

```
+-----------+--------------+--------------------+----------+-----------+---------+---------+--------------------+--------
------+---------+----------+
|       name|       country|              region|      date|avghumidity|avgtemp_c|avgvis_km|           condition|totalpre
cip_in|maxtemp_c|mintemp_c|
+-----------+--------------+--------------------+----------+-----------+---------+---------+--------------------+--------
------+---------+----------+
|     Mumbai|         India|         Maharashtra|2023-05-18|       67.0|     30.6|     10.0|               Sunny|
0.0|     34.5|     28.4|
|      Delhi|        Canada|             Ontario|2023-05-18|       60.0|     10.3|     10.0|               Sunny|
0.0|     19.5|      2.6|
|    Kolkata|         India|         West Bengal|2023-05-18|       66.0|     33.0|      9.9|Patchy light rain...|
0.23|     41.9|     28.9|
|     London|United Kingdom|City of London, G...|2023-05-18|       75.0|     14.1|     10.0|   Light rain shower|
0.06|     18.2|     10.6|
|    Chennai|         India|          Tamil Nadu|2023-05-18|       69.0|     32.2|     10.0|               Sunny|
0.0|     37.4|     29.3|
|  Bangalore|         India|           Karnataka|2023-05-18|       48.0|     29.5|     10.0|Patchy rain possible|
0.02|     35.9|     25.4|
|      Kochi|         India|              Kerala|2023-05-18|       80.0|     29.2|      9.8|Thundery outbreak...|
0.02|     33.9|     25.8|
|Gandhinagar|         India|             Gujarat|2023-05-18|       39.0|     34.8|     10.0|               Sunny|
```

```
import pandas as pd
```

```
df = pd.read_csv('/tmp/spark_output/daily_avg_weather_data.csv')
```

```
  FileNotFoundError: [Errno 2] No such file or directory: '/tmp/spark_output/daily_avg_weather_data.csv'
```

```
%sql
select name,date,condition_description,count(condition_description) as weather_status from daily_hourly_data group by
1,2,3
```

Table Visualization 1

	name	date	condition_description	weather_status
1	Kolkata	2023-05-22	Sunny	14
2	Bangalore	2023-05-22	Patchy light drizzle	2
3	Delhi	2023-05-21	Clear	6
4	Mumbai	2023-05-21	Sunny	12
5	Delhi	2023-05-21	Overcast	1
6	Amaravati	2023-05-22	Sunny	7
7	Amaravati	2023-05-21	Clear	10

306 rows

Table Visualization 1

	name	number_of_records
1	Bangalore	12
2	Kochi	12
3	Chennai	12
4	Shimla	12
5	London	12
6	Mumbai	13
7	Kolkata	12

11 rows