TITLE: Climate Resilience Monitor: End-to-End Analytics Pipeline

PROJECT TYPE: Capstone Project - Data Engineering & Business Intelligence

AUTHOR: Kamran Namazov

DATE: December 2025

1.Introduction

This capstone project focuses on analyzing climate, air quality, and extreme weather data for seven neighboring countries that are experiencing increasing heatwaves, pollution spikes, and seasonal storms.

Business Problem: Regional governments need to identify which cities are entering high-risk climate periods. The objective is to determine which factors such as temperature, AQI, and infrastructure vulnerability , predict economic damage and population exposure. This analysis enables decision-makers to prepare resources and prevent economic loss proactively.

To address this, raw daily climate data was collected from major cities to build a **"Climate Resilience Monitor."** The project is implemented in two stages: SQL for data engineering and Power BI for visual decision support.

This project utilizes a synthetic dataset generated via **Python** to simulate complex, real-world enterprise scenarios including null handling, data inconsistencies, and high-volume event logging.

The raw data is provided as CSV files for multiple countries (Azerbaijan, Georgia, Iran, Kazakhstan, Russia, Turkey, and Turkmenistan). Each file contains around 500 records with fields such as:

- Date, Country, City
- Temperature, Humidity, Precipitation
- Air Quality Index (AQI)
- Extreme weather event descriptions
- Population exposure
- Economic impact estimates
- Infrastructure vulnerability scores

The project is implemented in two main stages:

1. **SQL (data engineering and analysis)** – loading raw CSV files into SQL Server, cleaning and transforming the data, building an analysis-ready table, and performing analytical queries.
2. **Power BI (visualization)** – creating interactive dashboards on top of the cleaned SQL data.

2. Data Sources & Tools

Data sources:

- Multiple CSV files, one per country:
    - Azerbaijan_climate_500.csv
    - Georgia_climate_500.csv
    - Iran_climate_500.csv
    - Kazakhstan_climate_500.csv
    - Russia_climate_500.csv
    - Turkey_climate_500.csv
    - Turkmenistan_climate_500.csv
- Each file contains daily or periodic climate records with common columns (Record_ID, Date, Country, City, Temperature_C, Humidity_pct, Precipitation_mm, Air_Quality_Index_AQI, Extreme_Weather_Events, Population_Exposure, Economic_Impact_Estimate, Infrastructure_Vulnerability_Score, etc.).

Record ID	Unique identifier for each observation row.
Date / Country / City	The temporal and geospatial context of the record.
Temperature (°C)	Daily average temperature in Celsius.
Precipitation (mm)	Amount of rainfall or snowfall in millimeters.
Air Quality Index (AQI)	Daily air pollution level based on the standard AQI scale.
Extreme Weather Events	Categorical description of events (e.g., Drought, Storm, Heatwave) if recorded.
Biome_Type	Dominant ecological biome around the city (e.g., Urban, Steppe, Shrubland).
Population_Exposure	Estimated number of people directly affected by the daily climate conditions.
Economic_Impact_Estimate	Estimated financial loss caused by climate events on that day (Currency: USD).
Infrastructure_Vulnerability_Score	A risk score (0-100) indicating the fragility of local infrastructure against weather events.

Tools used:

- **SQL Server** (database name: CLIMATE) – for ETL (Extract–Transform–Load), data cleaning, modeling, and analytical queries.
- **SQL Server Management Studio (SSMS)** – to write and run SQL scripts.

- **Power BI** – to build interactive visualizations

3. Data Cleaning and ETL in SQL

The SQL part of the project implements a full ETL (Extract–Transform–Load) process:

3.1.Staging raw data

A staging table dbo.Climate_Staging is created with all columns defined as NVARCHAR (text). This allows raw CSV data to be loaded without worrying about formats or type errors.

SQL QUERY:

```
CREATE TABLE dbo.Climate_Staging (
    [Record_ID]                    NVARCHAR(50),
    [Date]                         NVARCHAR(50),
    [Country]                      NVARCHAR(100),
    [City]                         NVARCHAR(100),
    [Temperature_C]                NVARCHAR(50),
    [Humidity_pct]                 NVARCHAR(50),
    [Precipitation_mm]             NVARCHAR(50),
    [Air_Quality_Index_AQI]        NVARCHAR(50),
    [Extreme_Weather_Events]       NVARCHAR(100),
    [Climate_Classification]       NVARCHAR(50),
    [Climate_Zone]                 NVARCHAR(50),
    [Biome_Type]                   NVARCHAR(50),
    [Heat_Index]                   NVARCHAR(50),
    [Wind_Speed]                   NVARCHAR(50),
    [Wind_Direction]               NVARCHAR(50),
    [Season]                       NVARCHAR(50),
    [Population_Exposure]          NVARCHAR(50),
    [Economic_Impact_Estimate]     NVARCHAR(50),
    [Infrastructure_Vulnerability_Score] NVARCHAR(50)
);
```

*The CSV files for all countries are loaded into this staging table using **BULK INSERT:***

SQL QUERY:

TRUNCATE TABLE dbo.Climate_Staging;

BULK INSERT dbo.Climate_Staging

FROM 'C:\...\Azerbaijan_climate_500.csv'

WITH (FIRSTROW=2, FIELDTERMINATOR=',', ROWTERMINATOR='\n', CODEPAGE='65001', DATAFILETYPE='char');

-- Similar BULK INSERT commands for Georgia, Iran, Kazakhstan, Russia, Turkey, Turkmenistan

Climate_Staging_Table:

	Record_ID	Date	Country	City	Temperature_C	Humidity_pct	Precipitation_mm	Air_Quality_Index_AQI	Extreme_Weather_Events	Climate_Classification	Climate_Zone	Biome_Type	Heat_Index	Wind_Speed	Wind_Direct
1	AZ_0001	4/5/2024	Azerbaijan	Baku	NA	71	0	99	None	BSk	Semi-arid	Urban	6.1	5.3	129
2	AZ_0002	5/5/2024	Azerbaijan	Ganja	5.1	69	0	86	None	BSk	Mediterranean	Steppe	5.1	8.4	50
3	AZ_0003	6/5/2024	Azerbaijan	Sumqayit	12.0	75	0	79	None	BSk	Semi-arid	Steppe	12.0	2.6	193
4	AZ_0004	7/5/2024	Azerbaijan	Baku	6.9	90	NULL	92	None	BSk	Semi-arid	Shrubland	6.9	12.9	86
5	AZ_0005	8/5/2024	Azerbaijan	Ganja	10.9	78	3.8	74	None	NULL	NA	Shrubland	10.9	12.5	309
6	AZ_0006	9/5/2024	Azerbaijan	Sumqayit	6.6	61	0	108	None	Csa	Mediterranean	Shrubland	6.6	12.7	280
7	AZ_0007	10/5/2024	Azerbaijan	Baku	6.1	72	0.7	113	None	BSk	Mediterranean	Shrubland	6.1	2.2	42
8	AZ_0008	11/5/2024	Azerbaijan	Ganja	5.5	76	0	102	None	Csa	Semi-arid	Shrubland	5.5	9.8	NA
9	AZ_0009	12/5/2024	Azerbaijan	Sumqayit	9.8	94	0	128	None	Csa	Semi-arid	Shrubland	9.8	8.6	213
10	AZ_0010	13/5/2024	Azerbaijan	Baku	12.7	51	NULL	81	None	Csa	NA	Steppe	12.7	NA	288

3.2 Defining the Clean Schema

A template table dbo.Climate_Template is created with correct data types for all columns:

SQL QUERY:

```
CREATE TABLE dbo.Climate_Template (
    [Record_ID]              VARCHAR(20),
    [Date]                   DATE,
    [Country]                VARCHAR(50),
    [City]                   VARCHAR(100),
    [Temperature_C]          FLOAT,
    [Humidity_pct]           FLOAT,
    [Precipitation_mm]       FLOAT,
    [Air_Quality_Index_AQI]  INT,
```

```
    [Extreme_Weather_Events]          VARCHAR(100),
    [Climate_Classification]          VARCHAR(50),
    [Climate_Zone]                    VARCHAR(50),
    [Biome_Type]                      VARCHAR(50),
    [Heat_Index]                      FLOAT,
    [Wind_Speed]                      FLOAT,
    [Wind_Direction]                  VARCHAR(10),
    [Season]                          VARCHAR(20),
    [Population_Exposure]             INT,
    [Economic_Impact_Estimate]        FLOAT,
    [Infrastructure_Vulnerability_Score] INT
);
```

Per-country tables (e.g. dbo.Azerbaijan_climate_500, dbo.Georgia_climate_500, etc.) are then created from this template using SELECT INTO:

*SELECT * INTO dbo.Azerbaijan_climate_500*

FROM dbo.Climate_Template WHERE 1 = 0;

-- Similar commands for other countries

This ensures that all country tables share an identical, strongly-typed schema.

3.3 Cleaning and Type Conversion

Raw text data in the staging table is cleaned and converted into proper types in a temporary table #Cleaned. Key steps include:

- Trimming leading/trailing spaces (LTRIM, RTRIM)
- Converting empty strings to NULL (NULLIF)
- Removing thousands separators (, commas) from numeric text (REPLACE)
- Safely converting values to DATE, FLOAT, and INT using TRY_CONVERT

SQL QUERY:

```
SET DATEFORMAT dmy;
SELECT
```

```
    NULLIF(LTRIM(RTRIM(Record_ID)), '') AS Record_ID,
    TRY_CONVERT(DATE, NULLIF(LTRIM(RTRIM([Date])), '')) AS [Date],
    NULLIF(LTRIM(RTRIM(Country)), '') AS Country,
    NULLIF(LTRIM(RTRIM(City)), '') AS City,
    TRY_CONVERT(FLOAT, NULLIF(REPLACE(Temperature_C, ',', ''), '')) AS Temperature_C,
    TRY_CONVERT(FLOAT, NULLIF(REPLACE(Humidity_pct, ',', ''), '')) AS Humidity_pct,
    TRY_CONVERT(FLOAT, NULLIF(REPLACE(Precipitation_mm, ',', ''), '')) AS Precipitation_mm,
    TRY_CONVERT(INT, NULLIF(REPLACE(Air_Quality_Index_AQI, ',', ''), '')) AS
Air_Quality_Index_AQI,
    NULLIF(LTRIM(RTRIM(Extreme_Weather_Events)), '') AS Extreme_Weather_Events,
    NULLIF(LTRIM(RTRIM(Climate_Classification)), '') AS Climate_Classification,
    NULLIF(LTRIM(RTRIM(Climate_Zone)), '') AS Climate_Zone,
    NULLIF(LTRIM(RTRIM(Biome_Type)), '') AS Biome_Type,
    TRY_CONVERT(FLOAT, NULLIF(REPLACE(Heat_Index, ',', ''), '')) AS Heat_Index,
    TRY_CONVERT(FLOAT, NULLIF(REPLACE(Wind_Speed, ',', ''), '')) AS Wind_Speed,
    NULLIF(LTRIM(RTRIM(Wind_Direction)), '') AS Wind_Direction,
    NULLIF(LTRIM(RTRIM(Season)), '') AS Season,
    TRY_CONVERT(INT, NULLIF(REPLACE(Population_Exposure, ',', ''), '')) AS
Population_Exposure,
    TRY_CONVERT(FLOAT, NULLIF(REPLACE(Economic_Impact_Estimate, ',', ''), '')) AS
Economic_Impact_Estimate,
    TRY_CONVERT(INT, NULLIF(REPLACE(Infrastructure_Vulnerability_Score, ',', ''), '')) AS
Infrastructure_Vulnerability_Score
INTO #Cleaned
FROM dbo.Climate_Staging;
```

3.4 Removing Duplicate Records

Potential duplicate records are identified and removed based on Record_ID, Country, City, and Date using a window function:

SQL QUERY:

```
WITH Dupes AS (
```

```
   SELECT *, ROW_NUMBER() OVER (PARTITION BY Record_ID, Country, City, [Date] ORDER BY Record_ID) AS rn
   FROM #Cleaned
)
DELETE FROM Dupes WHERE rn > 1;
```

3.5 Loading into Per-Country and Merged Table

Cleaned rows from #Cleaned are inserted into each country table by filtering on Country. Finally, all country tables are merged into a single analysis-ready fact table **dbo.AllCountriesClimate_500**

SQL QUERY:

```
INSERT INTO dbo.Azerbaijan_climate_500
SELECT * FROM #Cleaned WHERE Country = 'Azerbaijan';
-- Similar INSERTs for other countries
SELECT *
INTO dbo.AllCountriesClimate_500
FROM (
   SELECT * FROM dbo.Azerbaijan_climate_500
   UNION ALL
   SELECT * FROM dbo.Georgia_climate_500
   UNION ALL
   SELECT * FROM dbo.Iran_climate_500
   UNION ALL
   SELECT * FROM dbo.Kazakhstan_climate_500
   UNION ALL
   SELECT * FROM dbo.Russia_climate_500
   UNION ALL
   SELECT * FROM dbo.Turkey_climate_500
   UNION ALL
   SELECT * FROM dbo.Turkmenistan_climate_500
```

) AS merged;

***dbo.AllCountriesClimate_500** table:*

	Record_ID	Date	Country	City	Temperature_C	Humidity_pct	Precipitation_mm	Air_Quality_Index_AQI	Extreme_Weather_Events	Climate_Classification	Climate_Zone	Biome_Type	Heat_Index	Wind_Speed	Wind_Direct
1	AZ_0002	2024-05-05	Azerbaijan	Ganja	5.1	69	0	86	None	BSk	Mediterranean	Steppe	5.1	8.4	50
2	AZ_0003	2024-05-06	Azerbaijan	Sumqayit	12	75	0	79	None	BSk	Semi-arid	Steppe	12	2.6	193
3	AZ_0005	2024-05-08	Azerbaijan	Ganja	10.9	78	3.8	74	None	NULL	NA	Shrubland	10.9	12.5	309
4	AZ_0006	2024-05-09	Azerbaijan	Sumqayit	6.6	61	0	108	None	Csa	Mediterranean	Shrubland	6.6	12.7	280
5	AZ_0007	2024-05-10	Azerbaijan	Baku	6.1	72	0.7	113	None	BSk	Mediterranean	Shrubland	6.1	2.2	42
6	AZ_0008	2024-05-11	Azerbaijan	Ganja	5.5	76	0	102	None	Csa	Semi-arid	Shrubland	5.5	9.8	NA
7	AZ_0009	2024-05-12	Azerbaijan	Sumqayit	9.8	94	0	128	None	Csa	Semi-arid	Shrubland	9.8	8.6	213
8	AZ_0011	2024-05-14	Azerbaijan	Ganja	8.9	52	2.3	100	None	Csa	Mediterranean	Shrubland	8.9	9.8	58
9	AZ_0012	2024-05-15	Azerbaijan	Sumqayit	5.9	92	0	122	None	Csa	Semi-arid	Steppe	5.9	14.9	339
10	AZ_0013	2024-05-16	Azerbaijan	Baku	13.8	83	2.3	93	None	BSk	Mediterranean	Steppe	13.8	8	92

- *The table AllCountriesClimate_500 is the main dataset used for all subsequent analysis*

4.Data Quality Checks

4.1 Missing Essential Values

Essential columns such as Record_ID, Date, Country, City, Temperature_C, Humidity_pct, Precipitation_mm, and Air_Quality_Index_AQI were checked for NULL values:

After building the unified table AllCountriesClimate_500, additional quality checks were performed to ensure that the data is reliable for analysis.

SQL QUERY:

```
SELECT COUNT(*) AS Nulls_InEssentialColumns

FROM dbo.AllCountriesClimate_500

WHERE Record_ID IS NULL

  OR [Date] IS NULL

  OR Country IS NULL

  OR City IS NULL

  OR Temperature_C IS NULL

  OR Humidity_pct IS NULL

  OR Precipitation_mm IS NULL

  OR Air_Quality_Index_AQI IS NULL;
```

Rows with missing values in any of these essential fields were removed:

SQL QUERY:

```
DELETE FROM dbo.AllCountriesClimate_500
WHERE Record_ID IS NULL
   OR [Date] IS NULL
   OR Country IS NULL
   OR City IS NULL
   OR Temperature_C IS NULL
   OR Humidity_pct IS NULL
   OR Precipitation_mm IS NULL
   OR Air_Quality_Index_AQI IS NULL;
```

	Nulls_InEssentialColumns
1	0

4.2 Duplicate Record IDs

A check for duplicate Record_ID values was performed to ensure uniqueness:

SQL QUERY:

```
SELECT Record_ID
FROM dbo.AllCountriesClimate_500
GROUP BY Record_ID
HAVING COUNT(*) > 1;
```

	Record_ID

This step ensures that each Record_ID identifies a unique record in the final dataset.

5. Analytical SQL Queries & Insights

With the cleaned and validated fact table AllCountriesClimate_500, several analytical queries were executed to explore climate trends, extreme weather events, air quality risks, and vulnerability.

5.1 Monthly Temperature Trends

SQL QUERY:

```
SELECT
    Country,
    YEAR([Date])          AS [Year],
    DATENAME(MONTH, [Date]) AS Month_Name,
    AVG(Temperature_C)     AS Avg_Temperature
FROM dbo.AllCountriesClimate_500
GROUP BY
    Country,
    YEAR([Date]),
    MONTH([Date]),
    DATENAME(MONTH, [Date])
ORDER BY
    Country,
    [Year],
    MONTH([Date]);
```

***Insight**: This query reveals seasonal temperature patterns and allows comparison of monthly averages across countries and years.*

	Country	Year	Month_Name	Avg_Temperature
1	Azerbaijan	2024	May	9.7304347826087
2	Azerbaijan	2024	June	30.4857142857143
3	Azerbaijan	2024	July	29.9551724137931
4	Azerbaijan	2024	August	30.788
5	Azerbaijan	2024	September	15.092
6	Azerbaijan	2024	October	15.0740740740741
7	Azerbaijan	2024	November	16.3407407407407
8	Azerbaijan	2024	December	3.525
9	Azerbaijan	2025	January	4.00689655172414
10	Azerbaijan	2025	February	4.71818181818182
18	Georgia	2024	May	7.45384615384615
19	Georgia	2024	June	27.9423076923077
20	Georgia	2024	July	27.1851851851852
21	Georgia	2024	August	28.1538461538461
22	Georgia	2024	September	12.8793103448276
23	Georgia	2024	October	11.6366666666667
24	Georgia	2024	November	11.7481481481481
25	Georgia	2024	December	0.625
26	Georgia	2025	January	0.976
27	Georgia	2025	February	0.052000000000...
28	Georgia	2025	March	8.00357142857143

5.2 Extreme Weather Events by Month

SQL QUERY:

```
SELECT
    DATENAME(MONTH, [Date]) AS Month_Name,
    COUNT(*)             AS Event_Count
```

```
FROM dbo.AllCountriesClimate_500

WHERE Extreme_Weather_Events IS NOT NULL

 AND Extreme_Weather_Events <> 'None'

GROUP BY

  DATENAME(MONTH, [Date]),

  MONTH([Date])

ORDER BY

  MONTH([Date]);
```

	Month_Name	Event_Count
1	January	50
2	February	40
3	March	28
4	April	24
5	May	53
6	June	97
7	July	98
8	August	95
9	September	32
10	October	28
11	November	25
12	December	55

Insight: *This shows which months have the highest number of extreme weather events, highlighting peak risk periods.*

5.3 Extreme Events by Country

The following query ranks countries by the number of recorded extreme weather events:

```
SELECT

  Country,

  COUNT(*) AS Event_Count

FROM dbo.AllCountriesClimate_500

WHERE Extreme_Weather_Events IS NOT NULL

 AND Extreme_Weather_Events <> 'None'

GROUP BY

  Country

ORDER BY
```

Event_Count DESC;

	Country	Event_Count
1	Turkmenistan	117
2	Iran	99
3	Russia	94
4	Kazakhstan	93
5	Turkey	89
6	Azerbaijan	76
7	Georgia	57

Insight: This helps identify which countries are most frequently affected by extreme weather in the dataset.

5.4 Air Quality Categories and Exposure

SQL QUERY:

```
SELECT
   CASE
      WHEN Air_Quality_Index_AQI <= 50 THEN 'Good (0-50)'
      WHEN Air_Quality_Index_AQI BETWEEN 51 AND 100 THEN 'Moderate (51-100)'
      WHEN Air_Quality_Index_AQI BETWEEN 101 AND 150 THEN 'Unhealthy for Sensitive (101-150)'
      WHEN Air_Quality_Index_AQI BETWEEN 151 AND 200 THEN 'Unhealthy (151-200)'
      WHEN Air_Quality_Index_AQI BETWEEN 201 AND 300 THEN 'Very Unhealthy (201-300)'
      ELSE 'Hazardous (>300)'
   END AS AQI_Category,
   COUNT(*) AS Records,
   SUM(Population_Exposure) AS Total_Population_Exposure,
   ROUND(AVG(Temperature_C), 1) AS Avg_Temperature,
   ROUND(AVG(Infrastructure_Vulnerability_Score), 0) AS Avg_Vulnerability
FROM dbo.AllCountriesClimate_500
WHERE Air_Quality_Index_AQI IS NOT NULL
GROUP BY
   CASE
```

```
    WHEN Air_Quality_Index_AQI <= 50 THEN 'Good (0-50)'

    WHEN Air_Quality_Index_AQI BETWEEN 51 AND 100 THEN 'Moderate (51-100)'

    WHEN Air_Quality_Index_AQI BETWEEN 101 AND 150 THEN 'Unhealthy for Sensitive (101-150)'

    WHEN Air_Quality_Index_AQI BETWEEN 151 AND 200 THEN 'Unhealthy (151-200)'

    WHEN Air_Quality_Index_AQI BETWEEN 201 AND 300 THEN 'Very Unhealthy (201-300)'

    ELSE 'Hazardous (>300)'

  END

ORDER BY

  Records DESC;
```

	AQI_Category	Records	Total_Population_Exposure	Avg_Temperature	Avg_Vulnerability
1	Moderate (51-100)	1920	613706987	15.3	53
2	Unhealthy for Sensitive (101-150)	1174	581576686	17	52
3	Good (0-50)	29	738051	17.5	50
4	Unhealthy (151-200)	1	891088	23.2	40

Insight: *This reveals how often each air-quality level occurs, how many people are exposed, and whether high AQI values tend to occur in more vulnerable environments.*

5.5 Seasonal Risk Analysis

SQL QUERY:

```
SELECT

  CASE

    WHEN Season IS NULL OR Season = 'NA' THEN 'Unknown'

    ELSE Season

  END AS Season,

  COUNT(*) AS Total_Records,

  SUM(CASE WHEN Extreme_Weather_Events IS NOT NULL

       AND Extreme_Weather_Events <> 'None'

      THEN 1 ELSE 0 END) AS Extreme_Weather_Count,
```

```
    ROUND(AVG(Temperature_C), 1) AS Avg_Temperature,
    ROUND(AVG(Precipitation_mm), 1) AS Avg_Precipitation,
    SUM(Population_Exposure) AS Total_Population_Exposure,
    SUM(Economic_Impact_Estimate) AS Total_Economic_Impact,
    ROUND(AVG(Infrastructure_Vulnerability_Score), 0) AS Avg_Vulnerability
FROM dbo.AllCountriesClimate_500
GROUP BY
    CASE
        WHEN Season IS NULL OR Season = 'NA' THEN 'Unknown'
        ELSE Season
    END
ORDER BY
    Extreme_Weather_Count DESC;
```

	Season	Total_Records	Extreme_Weather_Count	Avg_Temperature	Avg_Precipitation	Total_Population_Exposure	Total_Economic_Impact	Avg_Vulnerability
1	Summer	1114	279	32.5	3	450498088	135393018	53
2	Winter	554	140	-3.2	2.7	223405366	79527926	53
3	Spring	744	102	8.1	2.9	277730009	55878470	51
4	Autumn	643	83	12.7	2.8	219828409	32391528	52
5	Unknown	69	21	17.9	3.1	25450940	6778478	53

Insight: *This allows comparison of seasons by number of extreme events, average climate conditions, population exposure, and total economic impact.*

5.6 Biome-Level Risk Analysis

SQL QUERY:

```
SELECT
    CASE
        WHEN Biome_Type IS NULL OR Biome_Type = 'NA' THEN 'Unknown'
        ELSE Biome_Type
    END AS Biome_Type,
    COUNT(*) AS Total_Records,
    COUNT(DISTINCT Country + '|' + City) AS Locations_Affected,
    SUM(CASE WHEN Extreme_Weather_Events IS NOT NULL
```

```
        AND Extreme_Weather_Events <> 'None'
        THEN 1 ELSE 0 END) AS Extreme_Weather_Count,
    ROUND(AVG(Temperature_C), 1) AS Avg_Temperature,
    ROUND(AVG(Precipitation_mm), 1) AS Avg_Precipitation,
    SUM(Economic_Impact_Estimate) AS Total_Economic_Impact_Estimate,
    ROUND(AVG(Infrastructure_Vulnerability_Score), 0) AS Avg_Vulnerability
FROM dbo.AllCountriesClimate_500
GROUP BY
    CASE
        WHEN Biome_Type IS NULL OR Biome_Type = 'NA' THEN 'Unknown'
        ELSE Biome_Type
    END
ORDER BY
    Extreme_Weather_Count DESC;
```

	Biome_Type	Total_Records	Locations_Affected	Extreme_Weather_Count	Avg_Temperature	Avg_Precipitation	Total_Economic_Impact_Estimate	Avg_Vulnerability
1	Urban	885	18	170	15.4	3	72775345	53
2	Steppe	684	12	153	18.4	2.8	68802636	52
3	Desert	348	6	89	22.7	2.4	54241556	53
4	Forest	417	9	74	13.5	2.9	44322703	52
5	Shrubland	437	9	69	16.2	3.3	41495570	52
6	Taiga	134	3	33	5.5	2.7	13530371	53
7	Grassland	153	3	26	9	2.4	13334655	51
8	Unknown	66	21	11	14.6	3.6	1466584	51

Insight: *This shows which biomes (e.g., desert, forest, steppe, etc.) are most frequently affected by extreme weather, how widespread the impact is, and how high the economic and vulnerability levels are.*

6. Power BI Dashboard: Climate Resilience Monitor

6.1 Overview

Following the ETL and validation process in SQL, the dbo.AllCountriesClimate_500 table was connected to Microsoft Power BI to create the "Climate Resilience Monitor." This dashboard moves beyond descriptive reporting to provide a strategic Decision Support System.

The dashboard was designed with a specific focus on Resource Allocation, helping stakeholders identify high-risk infrastructure and inefficient economic spending caused by extreme weather events.

6.2 Data Modeling and DAX Logic

While SQL handled the data cleaning and categorization, DAX (Data Analysis Expressions) was used to create business-specific Key Performance Indicators (KPIs) that normalize the data for fair comparison across cities of different sizes.

A dedicated Date Table was created and linked to the main fact table to enable accurate time-intelligence analysis (monthly trends and drill-downs).

Key Measures Created:

Economic Severity ($/Capita):

Instead of looking at total cost (which biases towards large cities), this measure calculates the economic impact per person exposed.

Logic: DIVIDE(SUM(Economic_Impact), SUM(Population_Exposure))

Infrastructure Failure Risk %:

A critical risk metric that identifies the percentage of events occurring in areas with high vulnerability scores (>60).

Logic: Used CALCULATE with filter contexts and COALESCE to handle divide-by-zero errors when filters are applied.

Total Economic Impact:

A formatted currency measure to track total financial loss.

6.3 Dashboard Design & Layout

A custom Dark Mode JSON Theme was implemented to mimic professional monitoring software. This design choice reduces eye strain and allows color-coded alerts (Red/Green) to stand out effectively.

The layout follows a "Strategic Z-Pattern":

Top Row (KPIs): Immediate status check of the most critical numbers.

Left Sidebar (Filters): Granular control for the user.

Center (Analysis): Deep-dive tools for root cause and prioritization.

Bottom (Trends): Historical context.

6.4 Key Visualizations & Insights

A. The Executive HUD (KPI Cards)

The top row provides an instant health check of the monitored regions.

Features: Conditional formatting is applied to the values.

Temperature: Turns Red if Avg Temp > 25°C, Blue if < 10°C.

Infra Risk: Turns Red if the risk percentage exceeds 50%.

Insight: Enables stakeholders to spot anomalies (e.g., a sudden spike in Infrastructure Failure Risk) within seconds.

B. Root Cause Analysis (Decomposition Tree)

The central visual is Decomposition Tree that breaks down the Total Impact ($) metric.

Insight: Users can dynamically drill down to see that while "Droughts" may have the highest total cost, specific cities with "Critical" vulnerability are the primary drivers of those costs. This connects directly to SQL Query 5.3 and 5.6.

C. The Investment Matrix (Scatter Plot)

A Scatter Plot was constructed to act as a Risk Matrix, plotting Vulnerability Score (X-Axis) against Economic Severity (Y-Axis).

Features: Reference lines (averages) were added to divide the chart into four quadrants.

Insight: Cities falling in the Top-Right Quadrant represent "Critical Priority" targets—they have both weak infrastructure and high economic loss per capita. This visual answers the business question: "Where should we invest budget first?"

D. Air Quality Trends (Line Chart)

A time-series analysis linking to SQL Query 5.4.

Insight: Tracks Avg AQI over time to identify correlations between seasonal changes and air quality deterioration, helping predict future health risks.

6.5 User Experience (UX) & Interactivity

To facilitate deep exploration, several interactive elements were added:

Slicer Panel: A persistent sidebar allows users to filter by Country, City, Biome Type (using Dropdowns), and Season/Vulnerability (using Tile buttons).

Reset Functionality: A custom "Clear Slicers" button was scripted using Bookmarks to reset the dashboard to its default state instantly.

Cross-Filtering: Clicking on a specific bar in the Decomposition Tree automatically filters the Scatter Plot and KPIs, ensuring all data views remain synchronized.

6.6 Conclusion

This Power BI report completes the data pipeline. By visualizing the SQL-processed data, the project successfully transforms raw climate logs into actionable intelligence, enabling targeted interventions to improve climate resilience and optimize budget spending.