MySQL Database has been selected for solving the problem statements. The coding language used is Python. In order to Connect Python with MySQL, the library mysql-connector-python is utilised.  
  
**Script: P1\_DataModelling\_CreateTableScript.py**

Description:

This script contains the SQL Data Definition Language (DDL) statements to create a table for storing weather data records in a MySQL database.

Schema Design:

The table weather\_data includes the following fields:

id: A unique identifier for each record (Primary Key).

station\_id: An identifier for the weather station.

date: The date of the weather record.

max\_temp: The maximum temperature for that day (in tenths of a degree Celsius).

min\_temp: The minimum temperature for that day (in tenths of a degree Celsius).

precipitation: The amount of precipitation for that day (in tenths of a millimeter).  
  
  
  
**Script: P2\_IngestionScript.py**

Description:

This script reads weather data from text files, processes the data, and inserts it into the weather\_data table in the MySQL database. It ensures no duplicate records are inserted and logs the start and end times along with the number of records ingested.

Ingestion Process:

Connect to the MySQL database.

Read and process each file in the wx\_data directory.

Insert data into the weather\_data table.

Log the start and end times, and the number of records ingested.

A screenshot of a computer

Description automatically generated

**Script: P3\_CreateTableForDataAnalysisScript.py**

Description:

This script contains the SQL DDL statements to create a table for storing yearly weather statistics.

Schema Design:

The table weather\_stats includes the following fields:

id: A unique identifier for each record (Primary Key).

station\_id: An identifier for the weather station.

year: The year of the statistics.

avg\_max\_temp: The average maximum temperature for the year (in degrees Celsius).

avg\_min\_temp: The average minimum temperature for the year (in degrees Celsius).

total\_precipitation: The total precipitation for the year (in centimeters).

**Script: P3\_DataAnalysisScript.py**

Description:

This script calculates yearly statistics for each weather station and stores the results in the weather\_stats table. The calculations include average maximum temperature, average minimum temperature, and total accumulated precipitation.

**Data Analysis Process:**

1. Connect to the MySQL database.
2. Retrieve weather data grouped by station and year.
3. Calculate the required statistics.
4. Insert the calculated statistics into the weather\_stats table.

A screenshot of a computer

Description automatically generated

**Script: P4\_RestAPIScript.py**

This script sets up a Flask web application with endpoints to fetch weather data and statistics from the MySQL database. It allows filtering results by station ID and date/year through query parameters.

/api/weather: Returns ingested weather data.  
**Example URL:**

http://127.0.0.1:5000/api/weather?station\_id= USC00110072&date=1985-01-01

A screenshot of a computer

Description automatically generated

/api/weather/stats: Returns calculated statistics.

**Example URL:**

http://127.0.0.1:5000/api/weather/stats?station\_id=USC00110072&year= 1993

A screenshot of a computer

Description automatically generated