Weather Data ETL Project

Diego Iván Perea Montealegre [diego.perea@uao.edu.co](mailto:diego.perea@uao.edu.co)

Facultad de Ingeniería, Universidad Autónoma de Occidente

Cali, Valle del Cauca

**Introduction**

The analysis of meteorological data is essential for understanding climate patterns, predicting weather conditions, and studying the impact of climate on various areas such as agriculture, transportation, and disaster management. This synthetic dataset provides a valuable opportunity to research and develop models without the restrictions of accessing real data, facilitating experimentation and the improvement of analytical techniques.

The datset **Weather Data** of kaggle contains synthetic weather data generated for ten different locations, including New York, Los Angeles, Chicago, Houston, Phoenix, Philadelphia, San Antonio, San Diego, Dallas, and San Jose. The data includes information about temperature, humidity, precipitation, and wind speed, with 1 million data points generated for each parameter.

**Features:**

* **Location:** The city where the weather data was simulated.
* **Date\_Time:** The date and time when the weather data was recorded.
* **Temperature\_C:** The temperature in Celsius at the given location and time.
* **Humidity\_pct:** The humidity in percentage at the given location and time.
* **Precipitation\_mm:** The precipitation in millimeters at the given location and time.
* **Wind\_Speed\_kmh:** The wind speed in kilometers per hour at the given location and time.

**Additional Information:**

* **Variability and Complexity:** The dataset incorporates variability and complexity to simulate realistic weather patterns. For example, adjustments have been made to temperature and precipitation based on seasonal variations observed in certain locations. In New York, higher temperatures and precipitation are simulated during the summer months, while in Phoenix, lower temperatures and increased precipitation are simulated during the winter months.

**Importance and Applications**

1. **Development of Weather Prediction Models:** With parameter-based data, this dataset allows researchers and data scientists to train weather prediction models that can improve accuracy in estimating temperature, humidity, precipitation, and wind speed across different cities.
2. **Climate Studies and Trend Analysis:** By providing simulated weather data from various locations, this dataset enables the analysis of specific climate patterns in each region, the identification of long-term trends, and the evaluation of seasonal variations.
3. **Educational and Academic Use:** This dataset serves as an excellent tool for teaching data analysis, visualization, and modeling in meteorology. Students and educators can use it in courses related to data science, artificial intelligence, and applied statistics.
4. **Optimization in Climate-Dependent Sectors:** Industries such as agriculture, aviation, and logistics can utilize this data to simulate scenarios and develop adaptation strategies for climate variations, contributing to better planning and decision-making.
5. **Evaluation of Machine Learning Algorithms:** Thanks to the dataset's quantity and diversity, different machine learning approaches can be tested and compared for weather prediction, allowing for model validation before applying them to real-world data.

This synthetic dataset represents a versatile and valuable tool for research, education, and the practical application of climate analysis and prediction techniques. Its availability facilitates the development of innovative solutions without relying on access restrictions to real meteorological data.

**Objective**

The ETL (Extract, Transform, Load) process for visualizing this synthetic weather dataset would primarily aim to efficiently prepare the data to generate charts and visual analyses that help identify climate patterns, trends, and correlations between variables.

**Specific ETL Objectives:**

**Extraction (Extract):**

* Retrieve data from the original Kaggle dataset, ensuring data integrity.
* Load the data into an analysis environment such as Python (using Pandas), SQL, or a visualization tool like Power BI or Tableau.

**Transformation (Transform):**

* Data Cleaning:
  + Remove null or outlier values that could affect visualization accuracy.
  + Verify the consistency of measurement units (temperature in °C, wind speed in km/h, etc.).
* Format Conversion:
  + Convert the Date\_Time column into the appropriate date and time format for temporal analysis.
* Data Aggregation:
  + Group data by city to calculate daily, weekly, or monthly average values for temperature, humidity, precipitation, and wind speed.
* Generation of New Metrics:
  + Calculate moving averages or variation rates to observe climate trends.

**Final Project Delivery Steps**

1. Data Sources: Select one or more data sources (e.g., CSVs, APIs, databases).
2. Data Extraction: Use Python to extract data from the selected source and store it in a relational database.
3. Exploratory Data Analysis (EDA): Analyze the raw data to understand its structure and quality.
4. Read Raw Data: Retrieve the raw data from the staging area database using Python.
5. Data Transformation: Perform necessary transformations to create value and address the problem.
6. Merge Data (if needed): Combine different data sources through a merge task.
7. Load Processed Data: Store the transformed dataset back into the database.
8. Dashboard Creation: Retrieve data from the ETL pipeline database and create a dashboard using your preferred tool (e.g., Power BI, Looker Studio)

**Steps Project**

* **Phase1 :** Identification of the data problem or objective and dataset selection ,Data extraction or collection
* **Phase2 :** Data transformation , Data pre-analysis and visualization (EDA)
* **Phase3:** Data load in a SQL database , Presentation and Data story telling

**Phase 1**

The first phase involves extracting data from a source, in this case a .csv file, using Python and uploading the information to a database, which in this case is Supabase.

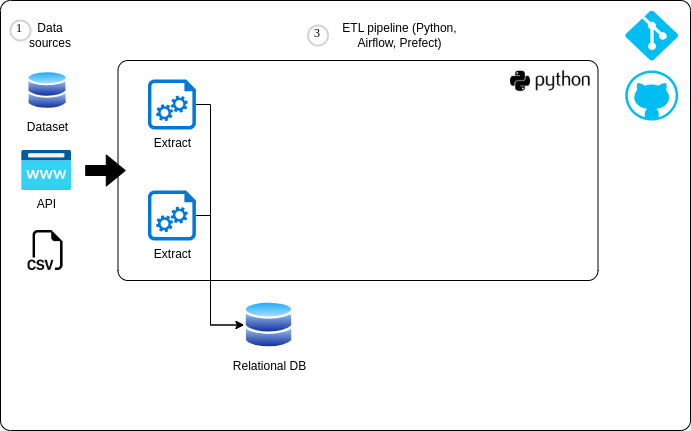


Figure 1. Diagram of Phase 1

**Steps for develop Phase 1:**

1. Create a database in Supabase. Supabase was selected because is an excellent choice for database management due to its ease of use, seamless setup, and cost-effectiveness. Unlike traditional databases that require complex installations and configurations, Supabase provides a fully managed, serverless PostgreSQL solution that can be accessed directly from the browser or through APIs. This eliminates the need for intricate setup processes, making it ideal for developers who want a hassle-free experience. Additionally, Supabase offers a generous free tier, allowing users to build and deploy applications without upfront costs. Its intuitive interface, real-time capabilities, and built-in authentication features further enhance its appeal, making it a powerful yet accessible choice for modern web and mobile applications.

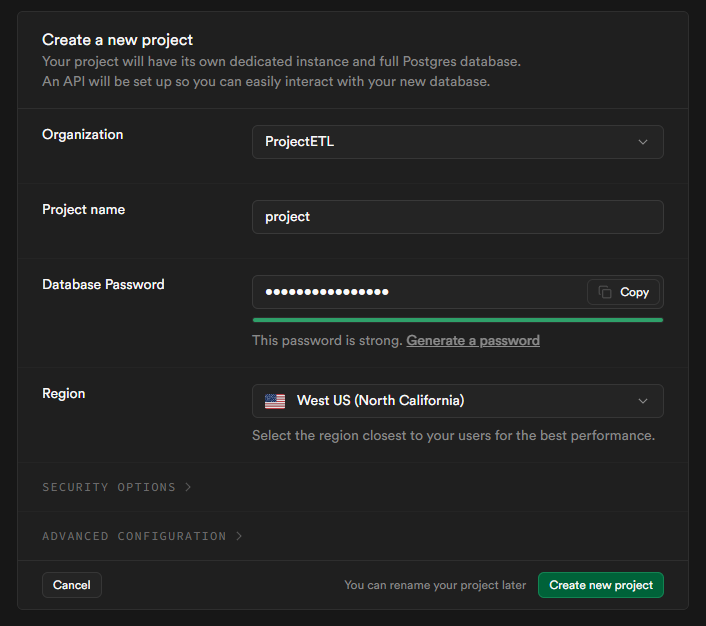


Figure 2. SQL Database Creation in Supabase

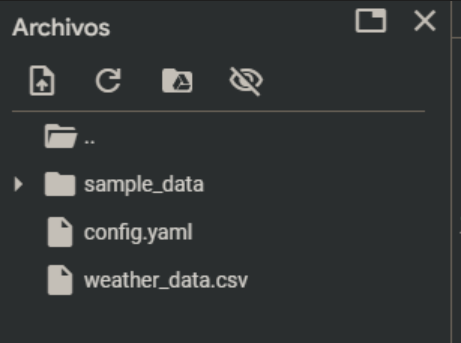
2. Connect the Supabase database and choose the connection, in this case, "Session pooler"



Figure 3. Conection with Session pooler

3. Upload data from the .csv to the database by following the file Phase1Load\001\_loadPhase1.ipynb

Download the **weather\_data.csv** file [1], create the **config.yaml** file following the steps, and upload it to Google Colab **.ipynb**.



4. View the uploaded data in the Supabase database. The next image shows evidence of the successful upload of the 11000 records handled in the previous step.

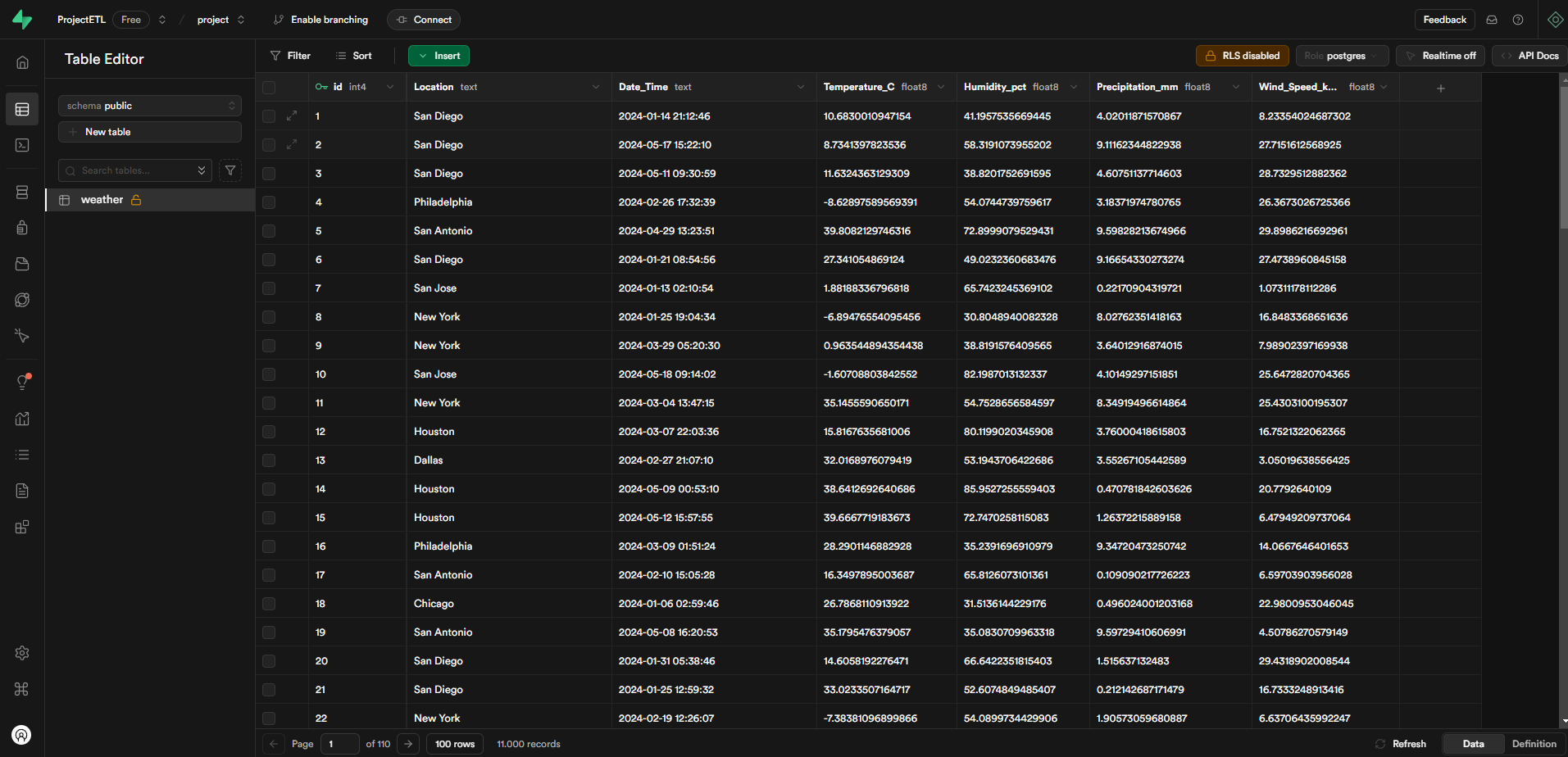


Figure 4. Uploaded Data in supabase database

**Phase 2**

The second phase performs the transformation and merging of the dataset uploaded to the DB.

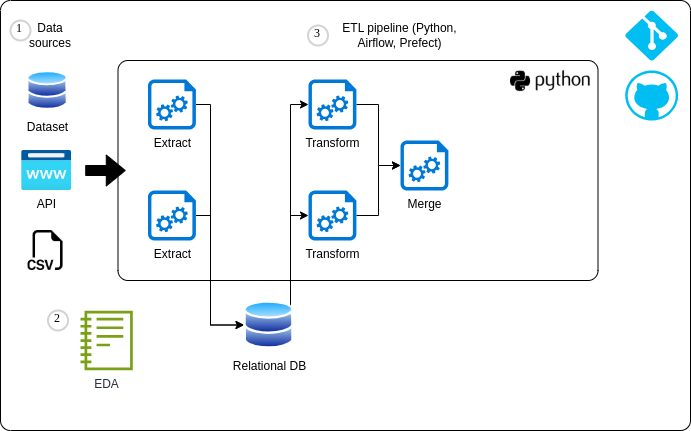


Figure 5. Diagram of Phase 2

The following transformations are performed:

\*  Extract year, month, day, hour, and minute from Date\_Time

\*  Convert Location to uppercase

\*  Normalize Temperature\_C to Fahrenheit

\*  Normalize Wind\_Speed\_kmh to mph

\*  Average temperature per location

Each of these transformations enhances the dataset's usability and improves its relevance for:

Extracting Year, Month, Day, Hour, and Minute from Date\_Time

* Reason: Breaking down timestamps into separate components allows for more granular time-based analysis. For example, it enables trend analysis by day, month, or year, which is useful for identifying seasonal patterns or hourly variations in weather conditions.
* Value: This facilitates time-series analysis, forecasting, and comparisons over different time scales.

Converting Location to Uppercase

* Reason: Standardizing location names eliminates inconsistencies caused by different letter cases (e.g., "San Diego" vs. "san diego"), ensuring that the same location is not treated as different entries.
* Value: Improves data integrity, prevents duplicate entries, and ensures consistency in filtering, grouping, and merging data.

Normalizing Temperature\_C to Fahrenheit

* Reason: Temperature is commonly reported in different units depending on the region. Converting Celsius to Fahrenheit ensures compatibility with audiences or models that require temperature in Fahrenheit.
* Value: Enhances accessibility for users in regions where Fahrenheit is the standard unit and allows for better cross-regional comparisons.

Normalizing Wind\_Speed\_kmh to mph

* Reason: Similar to temperature conversion, wind speed is often recorded in different units (km/h vs. mph). Converting it ensures consistency and usability across different datasets.
* Value: Enables seamless integration with other weather data sources and enhances the interpretability of wind speed information for users accustomed to mph.

Computing the Average Temperature per Location

* Reason: Aggregating temperature values provides insights into long-term trends and climate patterns for each location. This transformation helps smooth out short-term fluctuations and highlights the overall climate conditions in different regions.
* Value: Useful for climate monitoring, regional weather comparisons, and decision-making in sectors such as agriculture, energy, and urban planning.

**Importance of These Transformations**

These transformations enhance the dataset’s accuracy, consistency, and analytical potential. They enable:

* Better trend analysis (seasonal/weather patterns over time).
* Improved data consistency (avoiding duplicate or inconsistent values).
* Enhanced usability for different audiences (unit conversions).
* More effective aggregation and comparisons across locations.

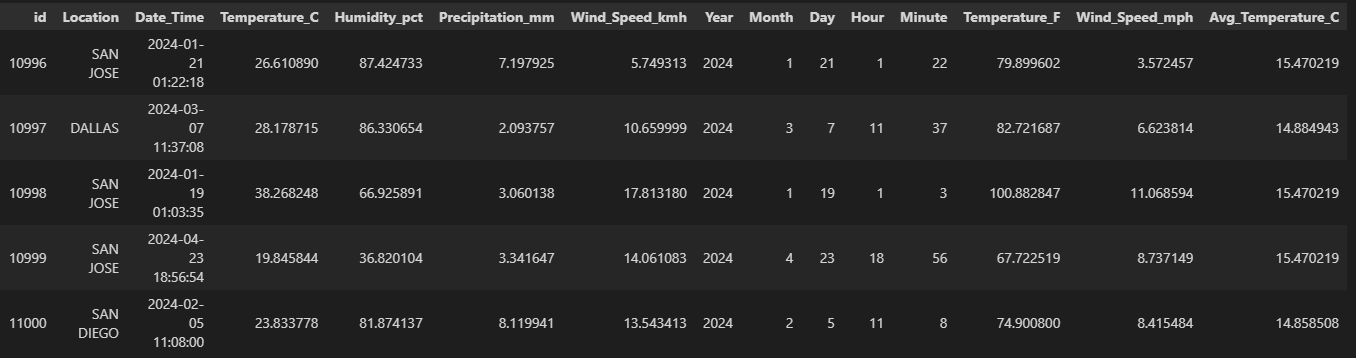


Figure 6. View of transformation

Upload to supabase

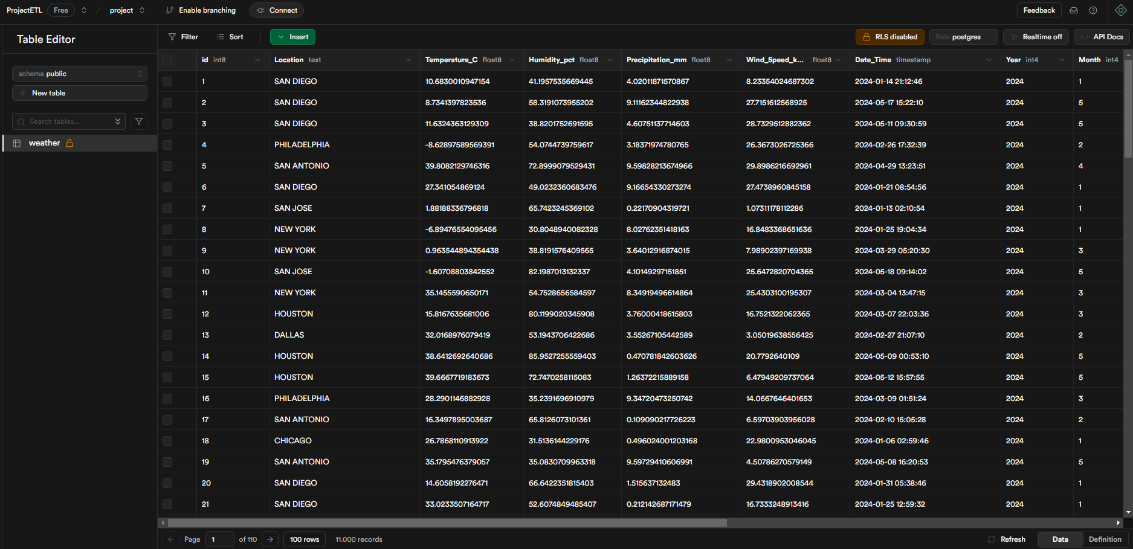


Figure 7. Transformation merged 1

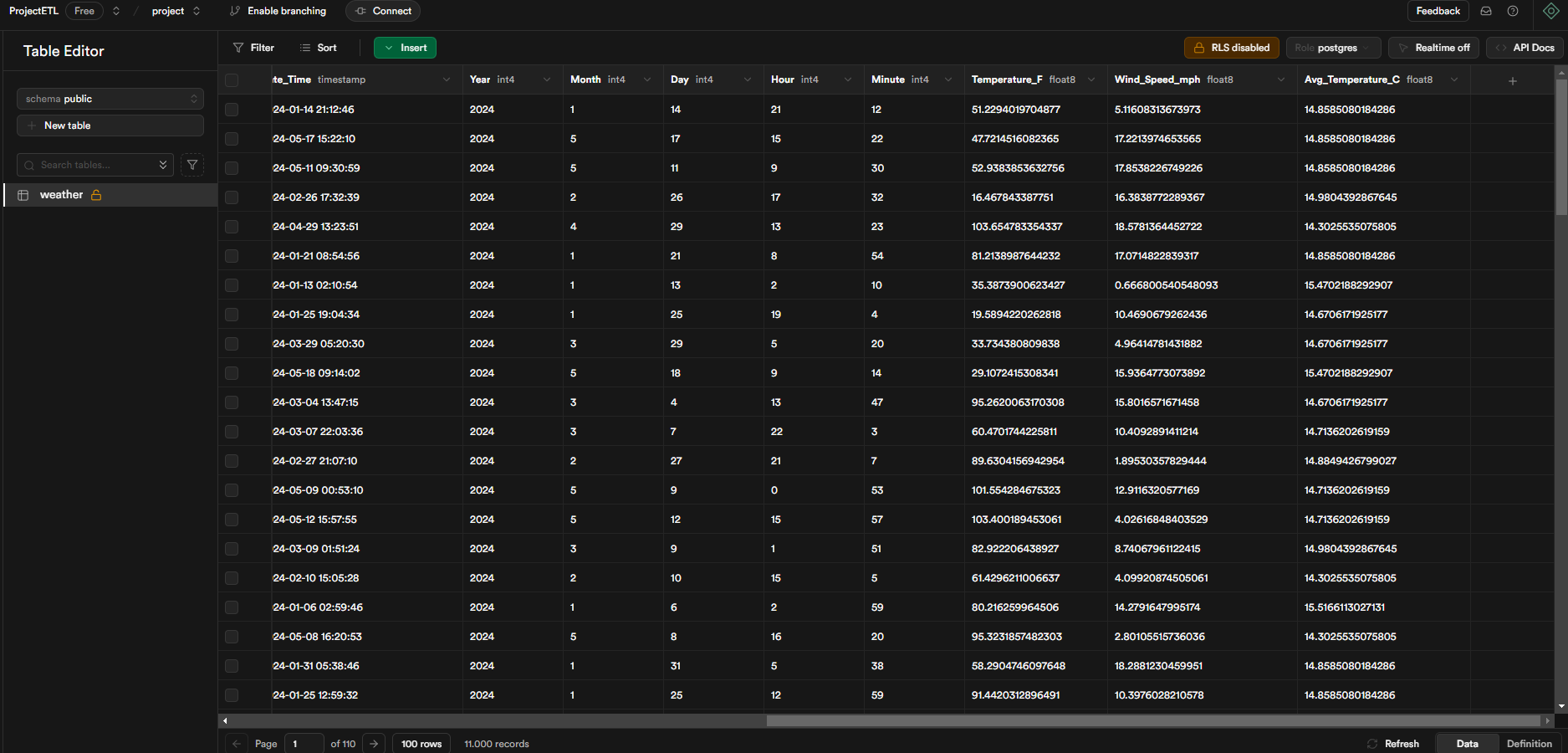


Figure 8. Transformation merged 2

**Phase 3**

The third phase performs the direct automation of the ETL part, where it is executed with a single command and the data set loaded into the database is displayed in a dashboard.

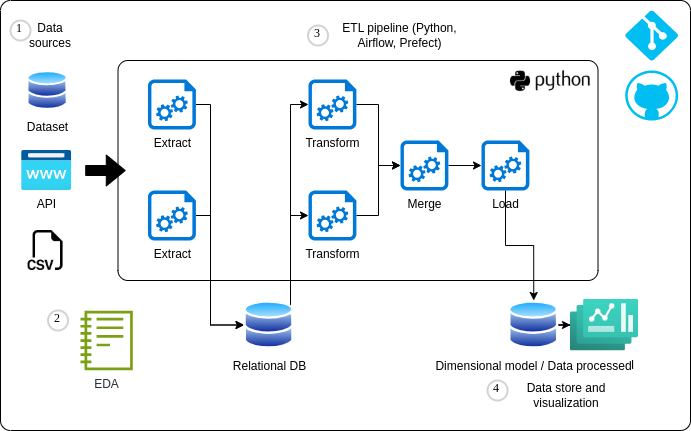


Figure 9. Diagram of Phase 3

Once the merge of phase two and the corresponding load in the database is done, the data visualization process is carried out, creating the dashboard to easily perform analysis of the data provided in order to determine actions or tasks based on the data obtained.

The dashboard was created with "Google Looker Studio" because of the ease of performing these processes and also because it is free and allows us to add databases such as PostgreSQL and others.

First, go to "<https://lookerstudio.google.com>" and register and log in. After that, create your respective project. To add data from our database such as Supabase which uses PostgreSQL, click on add data and search for PostgreSQL.

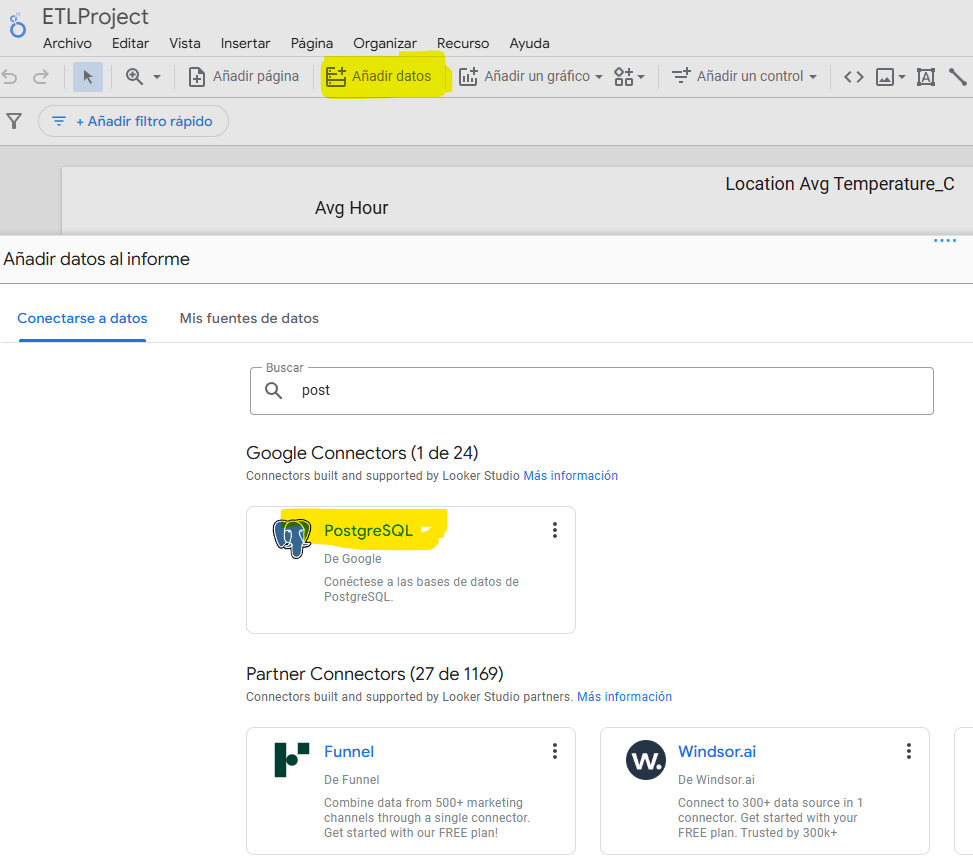


Figure 10. Process of adding data to the Looker Studio dashboard

After doing this, it requests the DB configuration in the following image. Since Supabase is being used, it should be done as follows: in databases enter "postgres" and for the port enter "5432". Where it asks for the host or IP, enter the URL, and for the username enter the user.

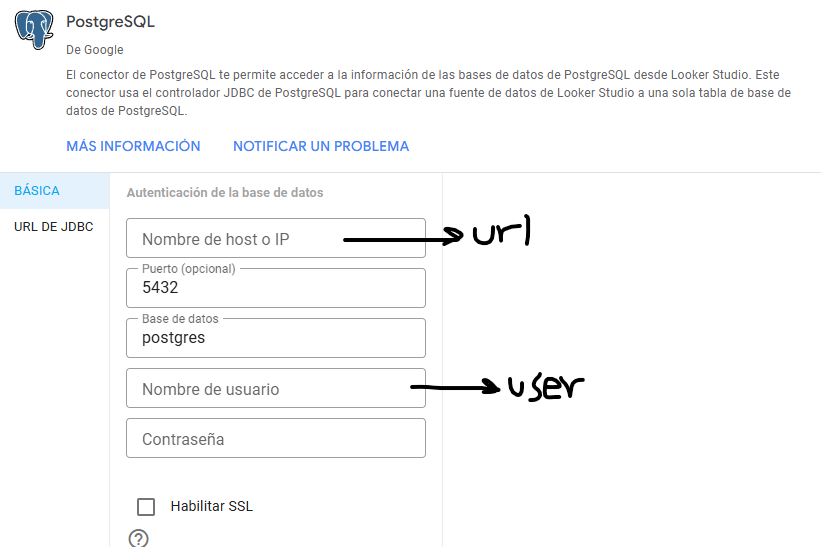


Figure 11. Process of adding DB data in Dashboard Looker Studio

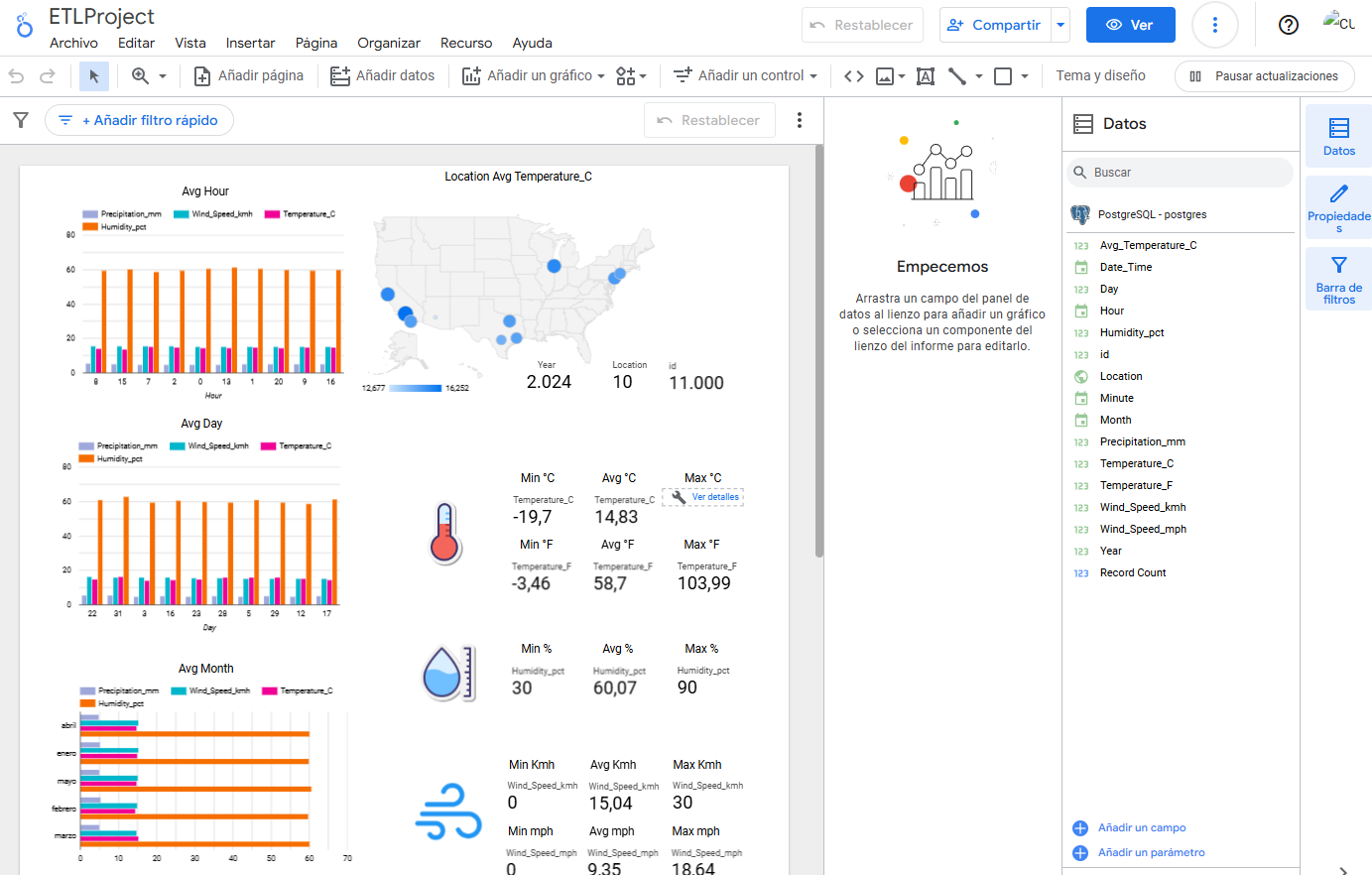
Once the successful connection with the DB is established, proceed to manipulate the data and create the dashboard with the different parameters and data obtained from the DB.

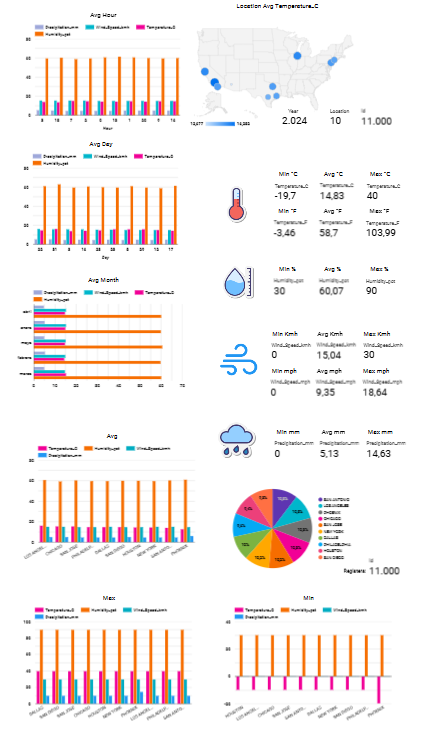
Figure 12. Creating the Looker Studio Dashboard

Figure 13. Dashboard complete looker studio

With the dashboard, we can perform different analyses regarding the data, such as multidimensional analysis of climate conditions and their interrelationships. We can study correlations between variables (such as temperature and humidity), temporal patterns (daily, seasonal, or annual variations), and geographic differences by comparing between cities. It is also possible to identify extreme events, calculate basic descriptive statistics for each variable, and apply advanced techniques such as time series analysis to predict future trends or detect climate anomalies.

To create the complete automation of the process for all phases of data extraction, transformation, and loading, with Python we create and structure a Python file where, through functions that correspond to each of the phases, it executes automatically in a direct way. This is in the file called "etl.py" which works, clarifying that our db is empty from the beginning, where the output provides evidence of the successful ETL process performed.

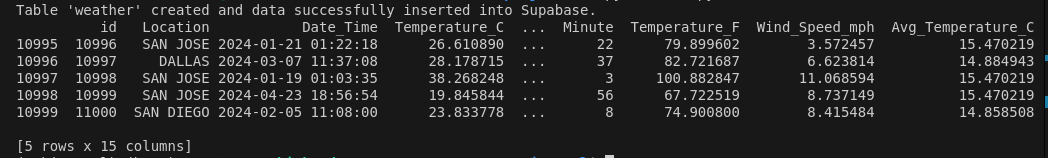


Figure 14. Successful Automatic Process direct from ETL

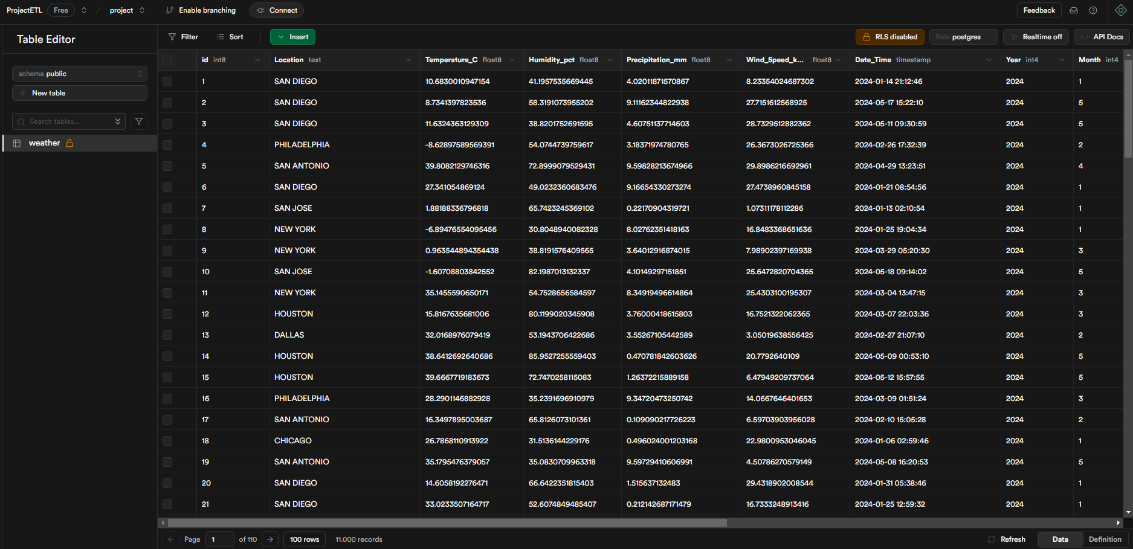


Figure 15. Supabase DB . Successful Automatic Process direct from ETLpart 1

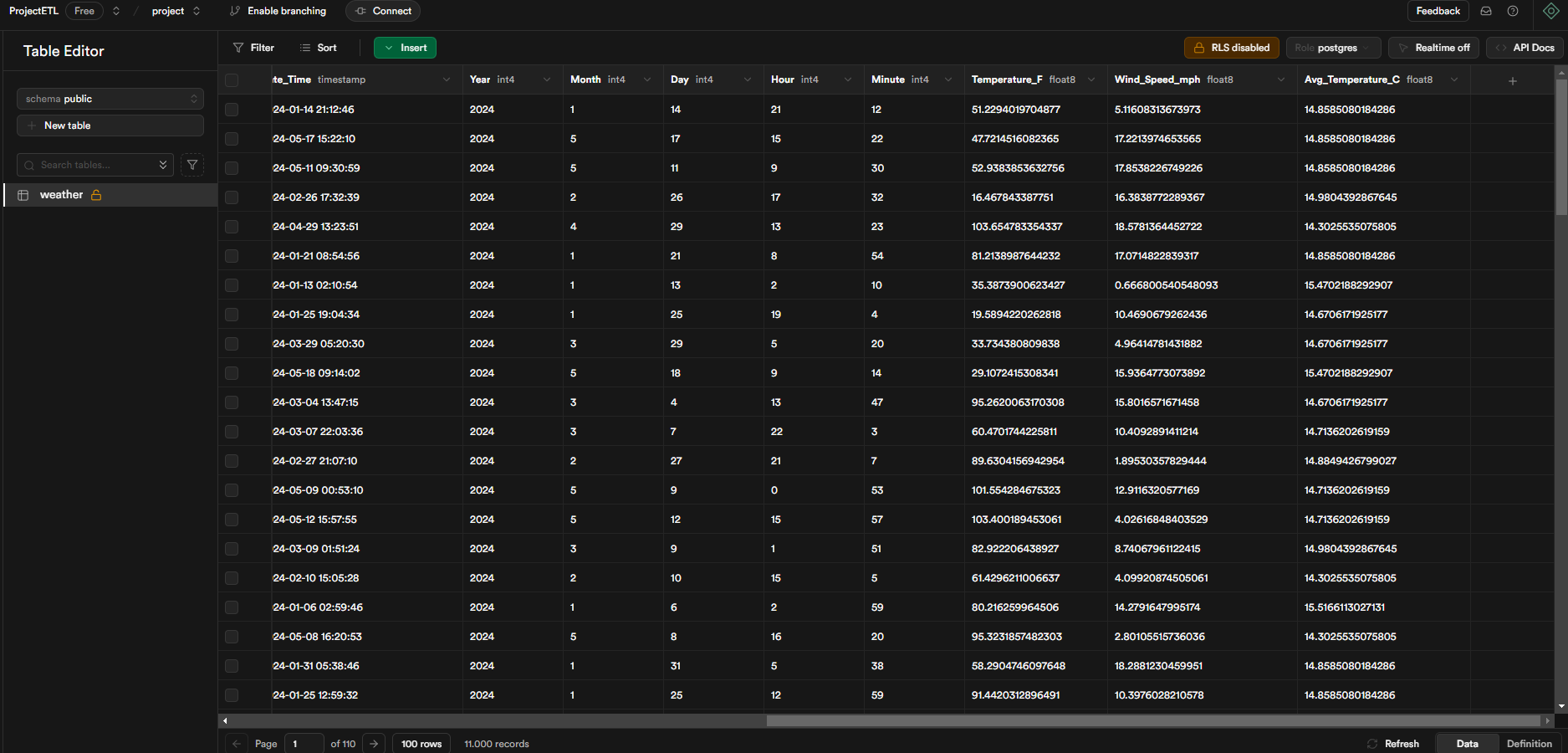


Figure 16. Supabase DB . Successful Automatic Process direct from ETL part 2

**References**

[1] Prasad22. (n.d.). *Weather data* [Dataset]. Kaggle. Retrieved from <https://www.kaggle.com/datasets/prasad22/weather-data>