E1- Schema Mapping – Milestone for "Purchased Energy (location based)"

DestinationTable Schema Overview **(data model explanation)**

User Inputs

AI schema mapping logic needs to be developed

Milestone Deliverables

List of Needed Resources

DestinationTable Schema Overview

Our Data Model includes the following tables:

a- List of **Dimention Tables**: (DE1_xx tables are dimension tables where each entry is unique and their IDs are used in the Fact table F1_xx)

1. **DE1_Scopes**: Contains fixed data. The 3 emission scopes defined by the GHG Protocol (Scope 1, Scope 2, Scope 3). This table will always remain the same, as the scopes are fixed.
2. **DE1_ActivityCategory**: Contains fixed categories of activities (e.g., purchased energy, business travel, waste disposal, etc.). This table will also remain the same, as the categories are predefined.
3. **DE1_ActivitySubCategory**: Contains fixed sub-categories of activities. This table will also always remain the same, as the sub-categories are predefined.
4. **DE1_ActivityEmissionSource**: Contains fixed emission sources (e.g., natural gas, diesel, air travel). This table will remain the same, as the sources are predefined.
5. **DE1_ActivityEmissionSourceProviders:** Contains a list of unique merchant names.
    - *E.g. if the user submits flight data, AI must populate this table by detecting the merchant from a possible field that has the airline-co data*.
6. **D_Country**: Contains a list of countries with their unique IDs (PK). The AI must populate this table by extracting country data from either the sourceTable (uploaded data) or the user input and assign unique IDs.
7. **DE1_Unit**: Contains fixed units of measurements. This table will also remain mostly the same, since we have fixed units of measurements.
8. **D_Company**: this is the Company/user that is submitting his data for sustainability reports. It contains company-related data (e.g., company name). The AI must populate this table based on either the source data or user's input.
9. **D_OrganizationalUnit**: Divisions/branches of a company. Based on the user input, AI needs to populate this table (using D_Company as a base table)
10. **D_Date**: Contains date-related data (e.g., year, month, day). The AI must extract and format date information from the sourceTable (or the submitted table).

- *E.g if the user submits flight data, AI must populate entries in this table by detecting the flight date from a possible field that has the date related data.*
- *Use OrderDate, ActivityDate, or BusinessTripDate (whichever is applicable) from the source.*

b- List of Dimention Tables: (DE1_xx tables are dimension tables where each entry is unique and their IDs are used in the Fact table F1_xx)

1. **FE1_EmissionActivityData**: Contains activity data (e.g., electricity bills, business trips) linked to emission scopes, categories, and sources. The AI must automatically determine the correct emission scope for each activity based on the GHG Protocol and populate this table accordingly.

   Important fields in this FACT table:

   - ***ConsumptionAmount***

     ConsumptionAmount varies by activity*:*

     i. ***Flights:*** *Calculate distance between departure and arrival cities (in KM).*

     ii. ***Hotels:*** *Number of* ***overnight stays****.*

     iii. ***Electricity:*** *kwh consumption*

     iv. ***Heating:*** *cubicMeter*

     v. *(More variations based on ActivityEmissionSource.)*

   - ***UnitID***

     UnitID Mapping Logic

     - For Expence-Base scenarios: Always use **UnitID = 13 or 14 (based on the currency**.)
     - For Consumption-Based Scenarios: Use **Activity-specific UnitID**:
       - Example: **Electricity → Unit "kWh", UnitID = 3**

- **ScopeID**

  Scope IDs are exactly the same IDs as the one connected to each Activity Category in DE1_ActivityCategory

  In our example: (Purchased Energy (location based)) it is always “2”

- **EmissionFactor**

  This field is generated from concatenation of **two** fields:

  1- ISO2Code (from D_Country)

  2- ActivityEmissionSourceName (from DE1_ActivityEmissionSource)

  ISO2Code+_+ActivityEmissionSourceName

  Example: DE_Green_Electricity

User Inputs (the existing py project can be resued/modified for this piece)

1. Initial user inputs (manually):

   Prompts already developed in the py project

   Company name, Country, Activity Category, Activty Subcategory, Reporting Year, Calcualtion Method, upload wizard

   prompts (that needs to be added in the streamlit app)

   - Prompt the user during ingestion to clarify whether the data belongs to:
     - A **single company/unit** or
     - **Multiple organisational units**

     Based on the input, AI needs to populate D_Company and D_OrdanizationalUnits

AI Logic to be developed

Develop an **AI-driven program** that automates the transformation of **raw source data** of ANY schema (some examples of sourceData are provided here for test purposes only) into structured, organized data aligned with our predefined schema (DestinationSchema.xlsx) and example destination tables (DestinationTables.xlsx). The AI must **intelligently map, sort, and validate** source data into the destination schema while flagging unresolved data for manual review.

Key Clarification:
The provided SourceData.xlsx is **only an example dataset** for development and testing. The AI solution must be **generalizable** and capable of handling **any dataset** with similar characteristics (e.g., unstructured or semi-structured data) and transforming it into the predefined destination schema, regardless of the source schema. The AI should not hard-code logic specific to the example dataset but instead derive patterns, relationships, and mappings dynamically.

Task Requirements

1. **Generalizability**:
    - The AI solution must **not hard-code logic specific to the example dataset** (test SourceData).
    - It should dynamically analyze and adapt to **any dataset** with similar characteristics, regardless of the source schema.
    - The solution must be capable of transforming, structuring, and organizing data into our predefined destination schema, even if the source dataset has a completely different schema.
2. **Data Mapping & Transformation**:
    - Analyze the source data and map it to the destination schema using AI/ML algorithms.
    - Account for variations in source data formats (e.g., date formats, categorical values, free-text fields).
    - Use DestinationTables.xlsx to infer relationships, data patterns, and validation rules (e.g., “CustomerID” must be unique).

3. **Dimension Table Management**:
    - Maintain fixed tables (Scopes, ActivityCategories, etc.)
    - Populate dynamic tables (Company, Country, Date, ActivityEmissionSourceProviders etc…)
    - For this task (purchased energy data _consumption & expense based), example of UnitIDs are:

3	kWh	(for electricity -consumption based)
12	cubic meters	(for heating -consumption based)
13	euros	(for expense based)
14	us dollars	(for expense based)

4. **Automated FACT Table Population**:
    - AI logic to map any source data fields to FE1_EmissionActivityData
    - Automatic detection of:
        - Consumptions
        - DateKey
        - Relevant emission factorsIDs
5. **Handling schema mapping functionalities for both:**
    - Expense based calculations
    - Consumption based calculations

6. **ID Management**:
    - Ensure proper handling of Primary Keys (PK) and Foreign Keys (FK).
    - For example, if there is an entry regarding an electricity consumption from Germany, the AI must:
        - Populate the D_Country table with Germany and its unique ID (PK).
        - Use that ID as a Foreign Key (FK) in the FE1_EmissionActivityData table.
    - Maintain consistency in ID assignment across all tables.

7. **Duplicate Handling**:
    - Detect duplicates in the source data (e.g., if the netto amount and tax amount for the same bill are saved in different rows).
    - Resolve duplicates by either:
        - Deleting the duplicate entry, or
        - Aggregating numeric values (e.g., summing the netto and tax amounts).
8. **Handling Unresolved Data**:
    - Flag records/fields that the AI cannot confidently map to the destination schema (e.g., missing required fields, ambiguous values, or untranslatable non-English entries).
    - Unresolved data with reasons (e.g., "No matching column for 'ProductCode' in target schema" or "Untranslatable French text in 'Comments' field").

Milestone Deliverables

1. **Code**:
    - A well-structured Python script/Jupyter Notebook (or equivalent) implementing the solution.
2. **Documentation**:
    - Technical report detailing methodology, challenges, and results.
3. **Outputs**:
    - Transformed data in the destination schema (exported as CSV/Excel) with all content in English.
    - UnresolvedData_Report.xlsx highlighting flagged records.

List of Needed Resources

DestinationSchema:

DestinationSchema.xlsx

DestinationTables:

DestinationTables_Fix Data Model.xlsx

Test SourceData:

Energy Source Data