Hurtowanie danych
Emisje gazów cieplarnianych

Autor: Aniela Brodziak

Kierunek studiów: Informatyka stosowana

Spis treści

Wstęp

Celem projektu było stworzenie hurtowni danych Emisji gazów cieplarnianych, zgodnie z zasadami metodologii Ralpha Kimballa. Podejście to zakłada budowę hurtowni danych w sposób zorientowany na dane biznesowe, wykorzystując modelowanie wymiarowo-faktowe, co umożliwia łatwe analizowanie i raportowanie danych.

Założenia projektu

Cel główny:

- Umożliwienie analiz emisji gazów cieplarnianych na poziomie krajów, źródeł emisji oraz typów gazów w określonych przedziałach czasowych.

Zakres danych:

- Dane z Gapminder o emisjach gazów takich jak CO2, CH4, N2O.
- Źródła emisji (np. transport, przemysł).
- Informacje o krajach, latach oraz ilościach emisji.

Etapy

Architektura

Tabele wymiarów:

- DimEmissionType: Przechowuje typy gazów cieplarnianych (np. CO2, CH4).
- DimEmissionSource: Przechowuje źródła emisji (np. przemysł, transport).
- DimCountry: Przechowuje nazwy krajów.
- DimCalendar: Przechowuje lata oraz dodatkowe informacje o czasie (np. czy rok jest przestępny).
- Skompresowane w trybie Page

Tabela faktów:

- FactEmissions: Przechowuje ilości emisji, połączone z wymiarami za pomocą kluczy obcych.
- Clustered Columnstore Index

Tworzenie bazy danych - SSMS

Wymiary i fakt

```sql
USE [master]
GO

-- Tworzenie bazy danych
CREATE DATABASE EmissionsDW -- Baza danych dla hurtowni danych dotyczących emisji
GO

USE [master]
GO

-- Ustawienie trybu odzyskiwania na prosty (SIMPLE) dla lepszej wydajności w hurtowni danych
ALTER DATABASE [EmissionsDW] SET RECOVERY SIMPLE WITH NO_WAIT
GO

USE [EmissionsDW]
GO

-- Tworzenie schematów dla porządkowania obiektów w bazie danych
CREATE SCHEMA [Emissions] -- Schemat dla tabel wymiarów i faktów
GO

CREATE SCHEMA [stg] -- Schemat dla tabel stagingowych (tymczasowych)
GO
```

Q 1 Stworzenie bazy, stworzenie schematów [stg], [Emissions]

```sql
/*
    Tworzenie tabel wymiarów
*/

-- TABELA EMISSIONTYPE (Wymiar typu emisji, np. CO2, CH4)
CREATE TABLE Emissions.DimEmissionType
(
    IDEmission INT IDENTITY(1,1) PRIMARY KEY NOT NULL,  -- Klucz główny
    EmissionType NVARCHAR(50) NOT NULL,                 -- Nazwa typu emisji (np. CO2, CH4)
    EmissionDescription AS (                            -- Kolumna obliczana: opis emisji
        CASE
            WHEN EmissionType = 'CO2' THEN 'Carbon dioxide'
            WHEN EmissionType = 'CH4' THEN 'Methane'
            WHEN EmissionType = 'N2O' THEN 'Nitrous oxide'
            ELSE 'Unknown gas'
        END
    ) PERSISTED -- Kolumna przechowywana
);
GO

-- Włącz kompresję strony dla tabeli wymiaru (oszczędność miejsca)
ALTER TABLE [Emissions].[DimEmissionType] REBUILD
WITH (DATA_COMPRESSION = PAGE);
GO
```

Q 2 Stworzenie wymiaru DimEmissionType

Wyrażenie case przyporządkowuje odpowiedni opis emisji do wzoru i dodaje do nowej kolumny.

```
-- TABELA EMISSIONSOURCE (Wymiar źródła emisji, np. transport, przemysł)
CREATE TABLE Emissions.DimEmissionSource
(
    IDEmissionSource INT IDENTITY(1,1) PRIMARY KEY NOT NULL, -- Klucz główny
    SourceName NVARCHAR(50) NOT NULL                         -- Nazwa źródła emisji
);
GO

ALTER TABLE [Emissions].[DimEmissionSource] REBUILD
WITH (DATA_COMPRESSION = PAGE);
GO

-- TABELA COUNTRY (Wymiar kraju, np. Polska, Niemcy)
CREATE TABLE Emissions.DimCountry
(
    IDCountry INT IDENTITY(1,1) PRIMARY KEY NOT NULL, -- Klucz główny
    CountryName NVARCHAR(50) NOT NULL                 -- Nazwa kraju
);
GO

ALTER TABLE [Emissions].[DimCountry] REBUILD
WITH (DATA_COMPRESSION = PAGE);
GO
```

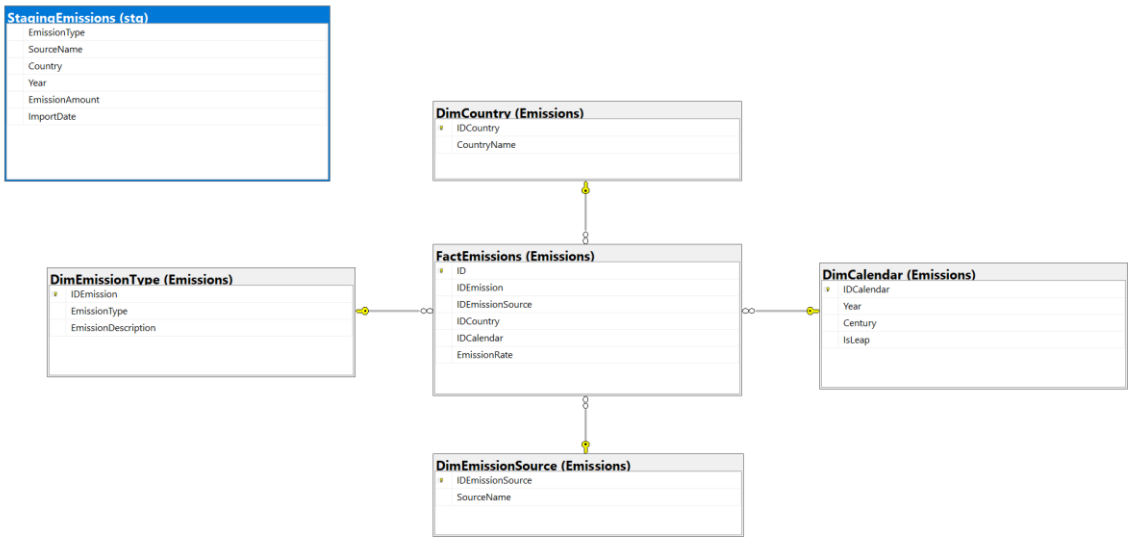
Q 3 Stworzenie wymiaru DimEmissionSource

```
-- TABELA CALENDAR (Wymiar czasu, rok i dodatkowe informacje o czasie)
CREATE TABLE Emissions.DimCalendar
(
    IDCalendar INT IDENTITY(1,1) PRIMARY KEY NOT NULL, -- Klucz główny
    Year INT NOT NULL,                      -- Rok
    Century AS (                                    -- Kolumna obliczana: wiek (np. 21 dla XXI wieku)
        CASE
            WHEN TRY_CAST(Year AS INT) IS NULL THEN NULL
            ELSE FLOOR(CAST(Year AS INT) / 100) + 1
        END
    ) PERSISTED,
    IsLeap AS (                                     -- Kolumna obliczana: czy rok jest przestępny (1 = tak, 0 = nie)
        CASE
            WHEN TRY_CAST(Year AS INT) IS NULL THEN NULL
            WHEN (CAST(Year AS INT) % 4 = 0 AND CAST(Year AS INT) % 100 != 0) OR (CAST(Year AS INT) % 400 = 0)
            THEN 1
            ELSE 0
        END
    ) PERSISTED
);
GO

ALTER TABLE [Emissions].[DimCalendar] REBUILD
WITH (DATA_COMPRESSION = PAGE);
GO
```

Q 4 Stworzenie wymiaru DimCalendar

```sql
/*
    Tworzenie tabeli faktu
*/

-- TABELA FACTEMISSIONS (Tabela faktu: przechowuje dane liczbowe dotyczące emisji)
CREATE TABLE [Emissions].[FactEmissions](
    [ID] INT IDENTITY(1,1) PRIMARY KEY NONCLUSTERED NOT NULL, -- Klucz główny (NONCLUSTERED, jawnie określony)
    [IDEmission] INT REFERENCES [Emissions].[DimEmissionType](IDEmission), -- Klucz obcy do wymiaru typu emisji
    [IDEmissionSource] INT REFERENCES [Emissions].[DimEmissionSource](IDEmissionSource), -- Klucz obcy do wymiaru źródła emisji
    [IDCountry] INT REFERENCES [Emissions].[DimCountry](IDCountry), -- Klucz obcy do wymiaru kraju
    [IDCalendar] INT REFERENCES [Emissions].[DimCalendar](IDCalendar), -- Klucz obcy do wymiaru kalendarza
    [EmissionRate] DECIMAL(10,4) NOT NULL -- Ilość emisji (liczba dziesiętna z precyzją do 4 miejsc po przecinku)
);

-- Dodanie klastrowanego indeksu kolumnowego dla optymalizacji analitycznej
CREATE CLUSTERED COLUMNSTORE INDEX EmissionsIdx ON Emissions.FactEmissions;
GO
```

Q 5 Stworzenie faktu



Q 6 Ostateczny diagram

Tabela stagingowa

- StagingEmissions: Tymczasowa tabela używana do załadowania i przekształcenia danych przed ich wstawieniem do tabel wymiarów i faktów.

```sql
-- TABELA STAGINGOWA (Do przechowywania danych wejściowych przed ich przetworzeniem)
CREATE TABLE [stg].[StagingEmissions] (
    EmissionType NVARCHAR(20), -- Typ emisji (np. CO2, CH4)
    SourceName NVARCHAR(30),   -- Nazwa źródła emisji (np. transport)
    Country NVARCHAR(50),      -- Nazwa kraju
    Year INT,                  -- Rok
    EmissionAmount DECIMAL(12,4), -- Ilość emisji
    ImportDate DATETIME DEFAULT GETDATE() -- Data zaimportowania danych
);
GO
```

Stworzenie procedur ładujących dane do wymiarów i faktu

```sql
/*
    Tworzenie procedur do ładowania danych
*/

-- Procedura do ładowania danych do wymiaru DimEmissionType
CREATE PROCEDURE [stg].[LoadDimension_DimEmissionType]
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO [Emissions].[DimEmissionType] (EmissionType)
    SELECT DISTINCT S.EmissionType
    FROM [stg].[StagingEmissions] S
    LEFT JOIN [Emissions].[DimEmissionType] ET
        ON S.EmissionType = ET.EmissionType
    WHERE ET.IDEmission IS NULL -- Wstaw tylko brakujące rekordy
    ORDER BY S.EmissionType ASC;
END;
GO
```

Q 7 Procedura dla DimEmissionType

```sql
-- Procedura do ładowania danych do wymiaru DimEmissionSource
CREATE PROCEDURE [stg].[LoadDimension_DimEmissionSource]
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO [Emissions].[DimEmissionSource] (SourceName)
    SELECT DISTINCT S.SourceName
    FROM [stg].[StagingEmissions] S
    LEFT JOIN [Emissions].[DimEmissionSource] ES
        ON S.SourceName = ES.SourceName
    WHERE ES.IDEmissionSource IS NULL
    ORDER BY S.SourceName ASC;
END;
GO
```

Q 8 Procedura dla DimEmissionSource

```sql
-- Procedura do ładowania danych do wymiaru DimCountry
CREATE PROCEDURE [stg].[LoadDimension_DimCountry]
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO [Emissions].[DimCountry] (CountryName)
    SELECT DISTINCT S.Country
    FROM [stg].[StagingEmissions] S
    LEFT JOIN [Emissions].[DimCountry] C
        ON S.Country = C.CountryName
    WHERE C.IDCountry IS NULL
    ORDER BY S.Country ASC;
END;
GO
```

Q 9 Procedura dla DimCountry

```sql
-- Procedura do ładowania danych do wymiaru DimCalendar
CREATE OR ALTER PROCEDURE [stg].[LoadDimension_DimCalendar]
AS
BEGIN
    SET NOCOUNT ON;

    -- Wstawienie unikalnych lat do tabeli DimCalendar
    INSERT INTO [Emissions].[DimCalendar] (Year)
    SELECT DISTINCT S.Year
    FROM [stg].[StagingEmissions] S
    LEFT JOIN [Emissions].[DimCalendar] DC
        ON S.Year = DC.Year -- Porównywanie bez zbędnego CAST
    WHERE DC.IDCalendar IS NULL
    ORDER BY S.Year ASC; -- Sortowanie w wynikach SELECT
END;
GO
```

Q 10 Procedura dla DimCalendar

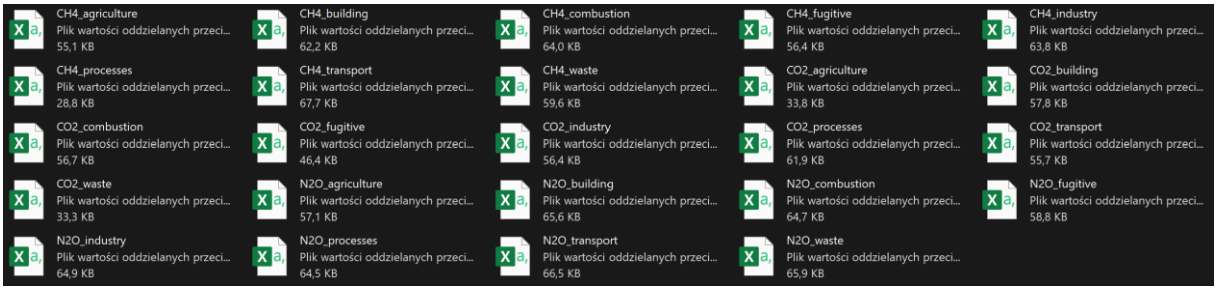
```sql
-- Procedura do ładowania danych do tabeli faktów FactEmissions
CREATE PROCEDURE [stg].[LoadDimension_FactEmissions]
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO [Emissions].[FactEmissions] (
        IDEmission, IDEmissionSource, IDCountry, IDCalendar, EmissionRate
    )
    SELECT
        DET.IDEmission,
        DES.IDEmissionSource,
        DC.IDCountry,
        DCAL.IDCalendar,
        SE.EmissionAmount
    FROM
        [stg].[StagingEmissions] SE
    INNER JOIN [Emissions].[DimEmissionType] DET
        ON SE.EmissionType = DET.EmissionType
    INNER JOIN [Emissions].[DimEmissionSource] DES
        ON SE.SourceName = DES.SourceName
    INNER JOIN [Emissions].[DimCountry] DC
        ON SE.Country = DC.CountryName
    INNER JOIN [Emissions].[DimCalendar] DCAL
        ON CAST(SE.Year AS NVARCHAR(50)) = DCAL.Year;
END;
GO
```

Q 11 Procedura dla FactEmissions

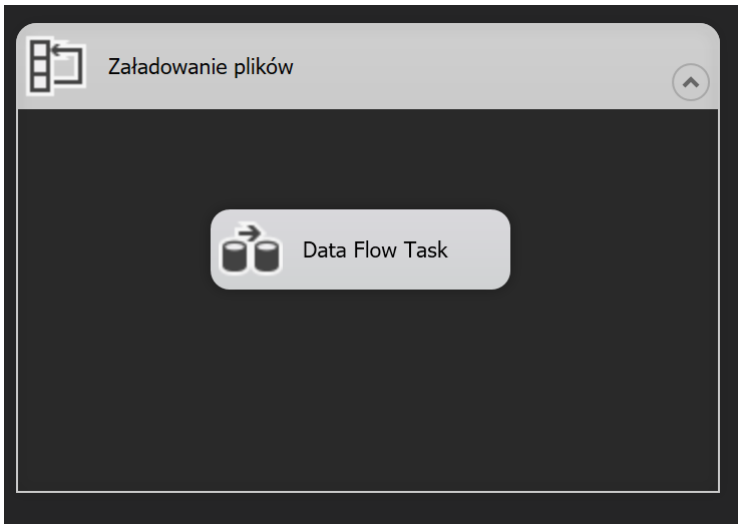
SSIS

Przygotowanie danych

Dane wejściowe zostały dostarczone w formie plików płaskich (CSV). Pliki te były załadowane do tabeli stagingowej za pomocą pakietu ETL w SSIS.

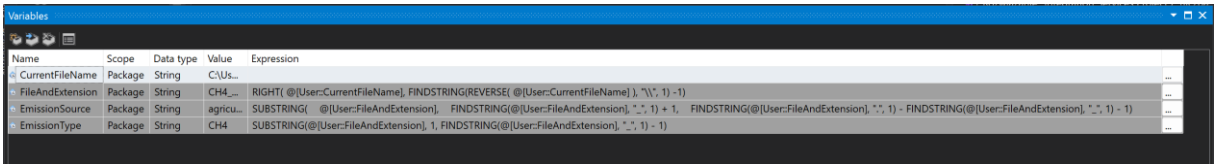


Rysunek 1 Pliki źródłowe



Rysunek 2 Control flow dla staging

Załadowanie plików za pomocą pętli foreach iterującej po plikach w przygotowanym katalogu.



Rysunek 3 Zmienne do iteracji po plikach

Dodano 4 zmienne:

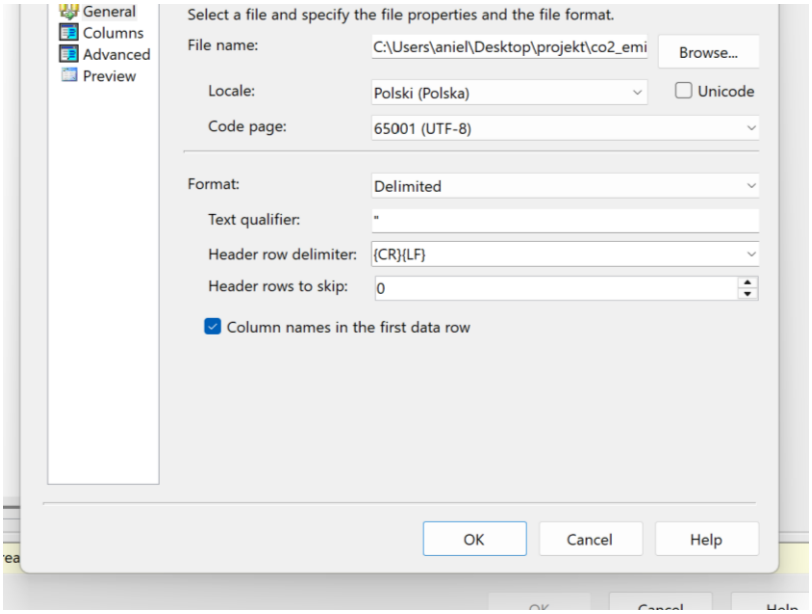
CurrentFileName – przechowująca pełną ścieżkę pliku

FileAndExtension – wyodrębniona nazwa pliku z jego rozszerzeniem

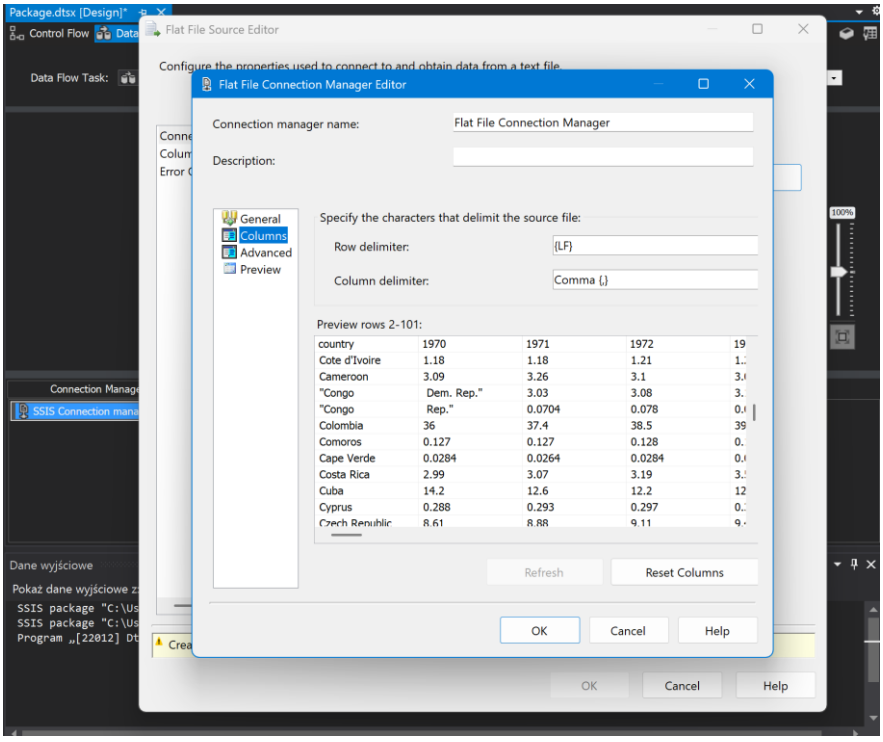
EmissionSource – źródło emisji, zapisane w nazwie pliku -> wyodrębnione na podstawie expression

EmissionType – typ emisji, zapisany w nazwie pliku -> wyodrębniony na podstawie expression

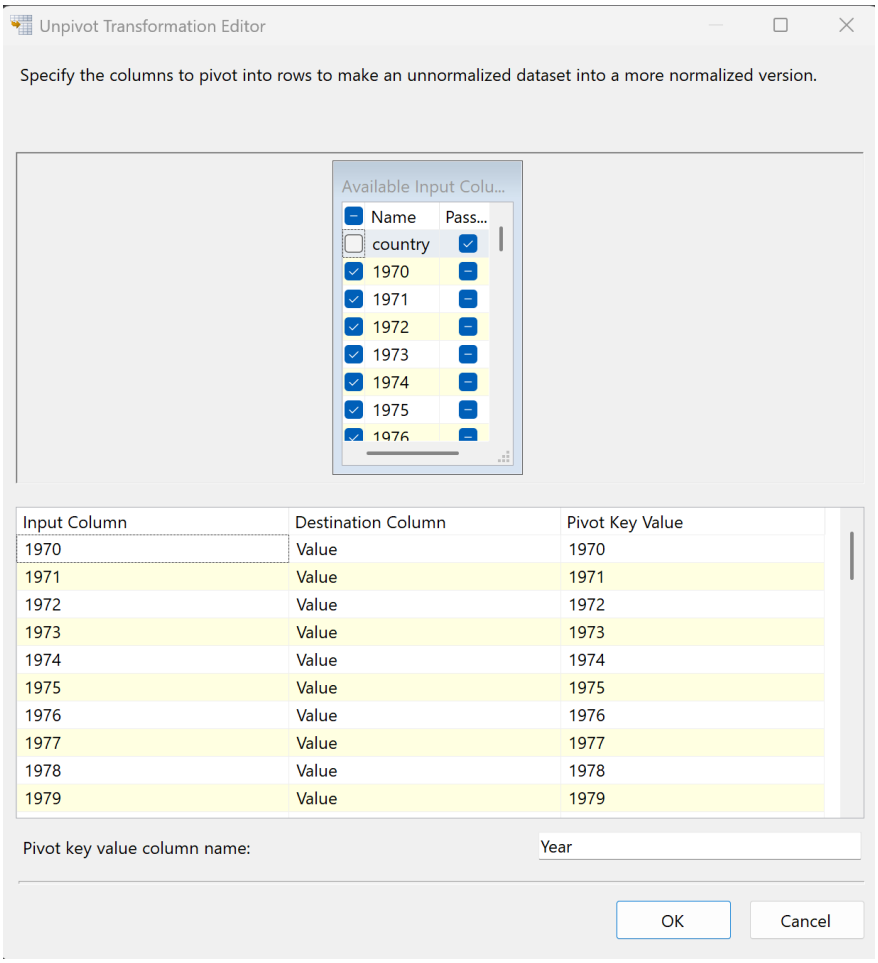
Transformacje danych



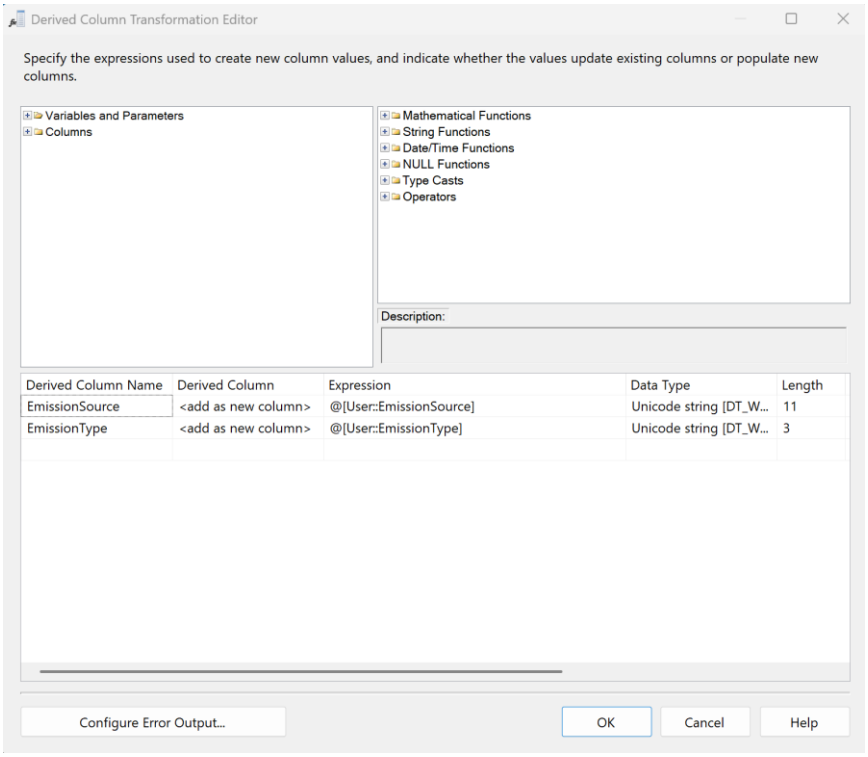
Rysunek 4 Ustawienia dla FlatFile



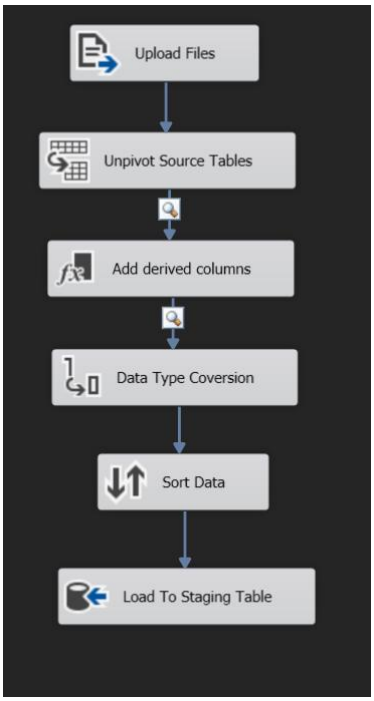
Rysunek 5 Ustawienia dla FlatFile



Rysunek 6 Unpivot dla tabeli przestawnej



Rysunek 7 Derived Column - dodanie kolumn [EmissionSource], [EmissionType]



Rysunek 8 Data Flow - dane źródłowe -> staging

- Upload Files – odpowiada za przesłanie plików CSV przy pomocy pętli foreach do flow.
- Unpivot Source Tables - przekształcenie danych z tabeli przestawnej
- Add Derived Columns - dodanie kolumn [Derived_EmissionSource] i [Derived_EmissionType] -> na podstawie zdefiniowanych wcześniej zmiennych zaczerpniętych z nazw plików

Derived Column Name	Derived Column	Expression	Data Type
Derived_EmissionSou...	<add as new column>	@[User::EmissionSource]	Unicode string [DT_W...
Derived_EmissionType	<add as new column>	@[User::EmissionType]	Unicode string [DT_W...

- Data Type Conversion - konwersja typów danych w kolumnach, zamiana formatu STRING na DT_DECIMAL -> dla Value (Scale = 4) oraz STRING na DT_I4 -> dla Year

Value	Converted_Emission...	decimal [DT_DECIMAL]			4	
Year	Converted_Year	four-byte signed integ...				

- Sort Data - posortowanie danych według wszystkich powstałych kolumn, usunięcie duplikatów
- Load to Staging Table – załadowanie danych źródłowych po transformacji do tabeli Stagingowej

Ładowanie danych do hurtowni:

Dane z tabeli stagingowej zostały załadowane do tabel wymiarów, a następnie do tabeli faktów.

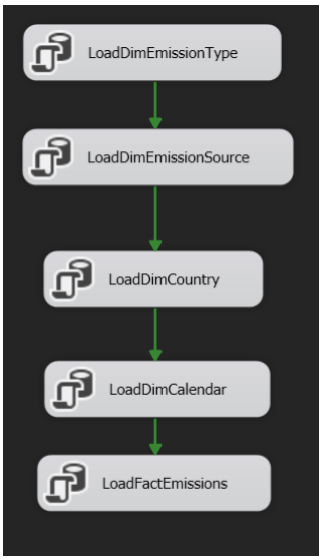
Proces ten był realizowany w dwóch oddzielnych pakietach SSIS:

Staging: Załadowanie danych ze źródła + transformacje do tabeli stagingowej.

Loading: Załadowanie danych do tabel.

Struktura pakietów SSIS

1. **Pakiet Staging**:
    - Załadowanie danych z plików wejściowych.
    - Przekształcenie danych (dodanie kolumn, konwersja typów, eliminacja duplikatów, sortowanie)
    - Wstawienie danych do tabeli stagingowej.
2. **Pakiet Loading**:
    - Ładowanie danych do tabel za pomocą EXECUTE SQL -> wywołanie procedur.
    - Wymiarów:
        - Emission Types.
        - Emission Sources.
        - Countries.
        - Calendar.
    - Faktu:
        - EmissionFact



Rysunek 9 Control Flow dla pakietu Loading

Execute SQL Task Editor

Configure the properties required to run SQL statements and stored procedures using the selected connection.

General
Parameter Mapping
Result Set
Expressions

General	
Name	**LoadDimEmissionType**
Description	**Execute SQL Task**
Options	
TimeOut	**0**
CodePage	**1250**
TypeConversionMode	**Allowed**
Result Set	
ResultSet	**None**
SQL Statement	
ConnectionType	**OLE DB**
Connection	**LocalHost.EmissionsDW**
SQLSourceType	**Direct input**
SQLStatement	**EXEC [stg].[LoadDimension_DimEmissionType];**
IsQueryStoredProcedure	False
BypassPrepare	**True**

Name
Specifies the name of the task.

Browse... Build Query... Parse Query

OK Cancel Help

Rysunek 10 Przykładowe ustawienia w Execute SQL Task dla pakietu Loading, LoadDimEmissionType

Podsumowanie

Stworzona baza danych emisji gazów cieplarnianych pozwala na:

Analizę emisji w różnych wymiarach:

Możliwość badania emisji gazów cieplarnianych (CO_2, CH_4, N_2O) w podziale na kraje, źródła emisji (np. transport, przemysł) oraz przedziały czasowe.

Centralizację i porządkowanie danych:

Wszystkie dane zostały zgromadzone w jednym miejscu, co eliminuje problemy związane z rozproszonymi źródłami i zapewnia ich spójność.

Przewidywanie i analizowanie trendów:

Historyczne dane pozwalają na identyfikowanie wzorców emisji oraz wspierają prognozowanie przyszłych wartości, co jest kluczowe dla strategii ochrony środowiska.

Elastyczność i skalowalność:

Rozwiązanie zostało zaprojektowane w sposób umożliwiający łatwe rozszerzenie o nowe źródła danych, typy emisji czy dodatkowe wymiary analizy.

Wsparcie w realizacji celów klimatycznych:

Baza dostarcza narzędzi do monitorowania emisji, co może wspierać kraje, firmy i organizacje w działaniach na rzecz ograniczenia emisji i realizacji polityk środowiskowych.