

Hurtownie Danych – laboratorium lista 5

Table of Contents

Zadanie 1. Przygotowanie powtarzalności procesu ETL	2
Zadanie 2. Wymiar Czasowy	2
Tworzenie tabeli:	3
Tworzenie tabeli pomocniczych:	4
Insertowanie danych:.....	5
Zadanie 3. Elementarne czyszczenie danych	6
Zadanie 4. Proces Extract – Transform – Load	7
Zadanie 5. ETL (prawie) bez SQLa	12
Insert Helpers:	12
DIM_TIME:.....	13
DIM_PRODUCT:.....	15
Wnioski:	17

Zadanie 1. Przygotowanie powtarzalności procesu ETL

Przygotować instrukcję usuwającą każdą z tabel utworzonych w trakcie pracy nad listą 4.

```

IF EXISTS (
    SELECT *
    FROM INFORMATION_SCHEMA.TABLES
    WHERE TABLE_SCHEMA = 'Czech' AND TABLE_NAME = 'FACT_SALES'
)
DROP TABLE Czech.FACT_SALES;
GO

IF EXISTS (
    SELECT *
    FROM INFORMATION_SCHEMA.TABLES
    WHERE TABLE_SCHEMA = 'Czech' AND TABLE_NAME = 'DIM_CUSTOMER'
)
DROP TABLE Czech.DIM_CUSTOMER;
GO

IF EXISTS (
    SELECT *
    FROM INFORMATION_SCHEMA.TABLES
    WHERE TABLE_SCHEMA = 'Czech' AND TABLE_NAME = 'DIM_PRODUCT'
)
DROP TABLE Czech.DIM_PRODUCT;
GO

IF EXISTS (
    SELECT *
    FROM INFORMATION_SCHEMA.TABLES
    WHERE TABLE_SCHEMA = 'Czech' AND TABLE_NAME = 'DIM SALESPERSON'
)
DROP TABLE Czech.DIM SALESPERSON;
GO

IF EXISTS (
    SELECT *
    FROM INFORMATION_SCHEMA.TABLES
    WHERE TABLE_SCHEMA = 'Czech' AND TABLE_NAME = 'DIM_TIME'
)
DROP TABLE Czech.DIM_TIME;
GO

```

Zadanie 2. Wymiar Czasowy

Przygotować wymiar czasowy: utworzyć i wypełnić danymi tabelę DIM_TIME. Tabela DIM_TIME powinna być tabelą zawierającą wymiar czasowy (klucze obce do tej tabeli znajdują się w tabeli faktów). Tabela DIM_TIME powinna zawierać następujące kolumny:

- PK_TIME (klucz główny – liczba całkowita postaci yyyymmdd – format taki sam jak kolumn OrderDate, ShipDate)
- Rok
- Kwartał
- Miesiąc

- Miesiąc słownie (wykorzystać tabelę pomocniczą z 12 rekordami dokonać odpowiedniego złączenia)
- Dzień tygodnia słownie (wykorzystać tabelę pomocniczą z 7 rekordami dokonać odpowiedniego złączenia)
- Dzień miesiąca

Tworzenie tabeli:

```
|CREATE TABLE Czech.DIM_TIME (  
    PK_TIME INT, -- np. 20240421  
    Rok INT,  
    Kwartał INT,  
    Miesiąc INT,  
    Miesiąc_Slow NVARCHAR(20),  
    Dzień_Tygodnia_Slow NVARCHAR(20),  
    Dzień_Miesiąca INT  
    CONSTRAINT PK_DIM_TIME PRIMARY KEY (PK_TIME)  
);
```

Tworzenie tabeli pomocniczych:

```

IF OBJECT_ID('tempdb..#Miesiace') IS NOT NULL
    DROP TABLE #Miesiace;

IF OBJECT_ID('tempdb..#DniTygodnia') IS NOT NULL
    DROP TABLE #DniTygodnia;

CREATE TABLE #Miesiace (
    Miesiac INT PRIMARY KEY,
    Miesiac_Slow NVARCHAR(20)
);

INSERT INTO #Miesiace (Miesiac, Miesiac_Slow)
VALUES
(1, 'Styczeń'), (2, 'Luty'), (3, 'Marzec'), (4, 'Kwiecień'),
(5, 'Maj'), (6, 'Czerwiec'), (7, 'Lipiec'), (8, 'Sierpień'),
(9, 'Wrzesień'), (10, 'Październik'), (11, 'Listopad'), (12, 'Grudzień');

CREATE TABLE #DniTygodnia (
    Dzień_Tygodnia INT PRIMARY KEY,
    Dzień_Tygodnia_Slow NVARCHAR(20)
);

INSERT INTO #DniTygodnia (Dzień_Tygodnia, Dzień_Tygodnia_Slow)
VALUES
(1, 'Niedziela'), (2, 'Poniedziałek'), (3, 'Wtorek'), (4, 'Środa'),
(5, 'Czwartek'), (6, 'Piątek'), (7, 'Sobota');

```

Insertowanie danych:

```

WITH CTE_Dates AS (
    SELECT CAST('20110531' AS DATE) AS D
    UNION ALL
    SELECT DATEADD(DAY, 1, D)
    FROM CTE_Dates
    WHERE D < '20141231'
)
INSERT INTO Czech.DIM_TIME
(PK_TIME, Rok, Kwartal, Miesiac, Miesiac_Slow, Dzień_Tygodnia_Slow, Dzień_Miesiaca)
SELECT
    CONVERT(INT, FORMAT(D, 'yyyyMMdd')) AS PK_TIME,
    DATEPART(YEAR, D) AS Rok,
    DATEPART(QUARTER, D) AS Kwartal,
    DATEPART(MONTH, D) AS Miesiac,
    m.Miesiac_Slow,
    dt.Dzień_Tygodnia_Slow,
    DATEPART(DAY, D) AS Dzień_Miesiaca
FROM CTE_Dates
JOIN #Miesiace m ON DATEPART(MONTH, D) = m.Miesiac
JOIN #DniTygodnia dt ON DATEPART(WEEKDAY, D) = dt.Dzień_Tygodnia
OPTION (MAXRECURSION 10000);

```

	PK_TIME	Rok	Kwartal	Miesiac	Miesiac_Slow	Dzień_Tygodnia_Slow	Dzień_Miesiaca
1	20110531	2011	2	5	Maj	Wtorek	31
2	20110601	2011	2	6	Czerwiec	Sroda	1
3	20110602	2011	2	6	Czerwiec	Czwartek	2
4	20110603	2011	2	6	Czerwiec	Piatek	3
5	20110604	2011	2	6	Czerwiec	Sobota	4
6	20110605	2011	2	6	Czerwiec	Niedziela	5
7	20110606	2011	2	6	Czerwiec	Poniedzialek	6
8	20110607	2011	2	6	Czerwiec	Wtorek	7
9	20110608	2011	2	6	Czerwiec	Sroda	8
10	20110609	2011	2	6	Czerwiec	Czwartek	9
11	20110610	2011	2	6	Czerwiec	Piatek	10
12	20110611	2011	2	6	Czerwiec	Sobota	11
13	20110612	2011	2	6	Czerwiec	Niedziela	12
14	20110613	2011	2	6	Czerwiec	Poniedzialek	13
15	20110614	2011	2	6	Czerwiec	Wtorek	14

Zadanie 3. Elementarne czyszczenie danych

Zamienić wszystkie wartości NULL:

- w kolumnie Color (tabela DIM_PRODUCT) na „Unknown”,
- w kolumnie SubCategoryName (tabela DIM_PRODUCT) na „Unknown”,
- w kolumnie CountryRegionCode na 000,
- w kolumnie Group na „Unknown”.

```
--zadanie 3. lista 5
]UPDATE Czech.DIM_PRODUCT
SET Color = 'Unknown'
WHERE Color IS NULL;

]UPDATE Czech.DIM_PRODUCT
SET SubCategoryName = 'Unknown'
WHERE SubCategoryName IS NULL;

]UPDATE Czech.DIM_CUSTOMER
SET CountryRegionCode = '000'
WHERE CountryRegionCode IS NULL;

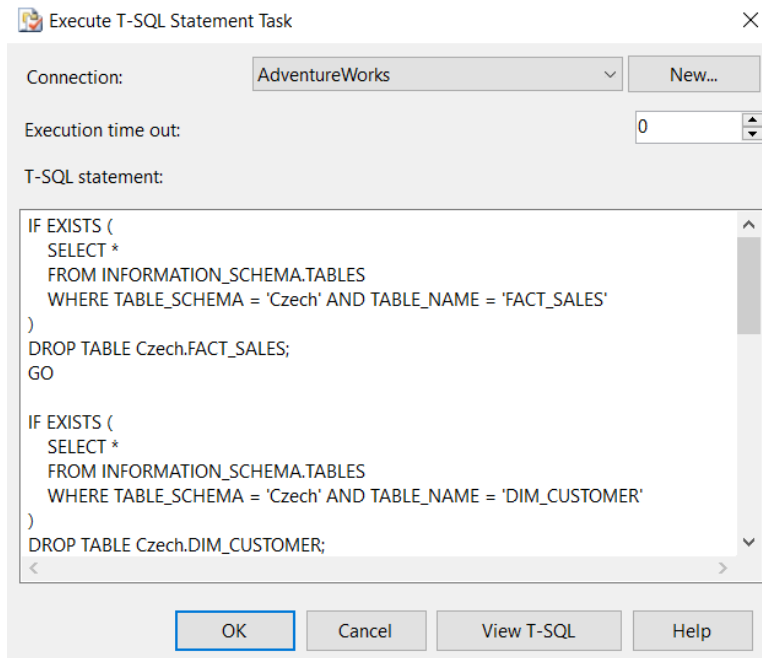
]UPDATE Czech.DIM_CUSTOMER
SET [Group] = 'Unknown'
WHERE [Group] IS NULL;
```

Zadanie 4. Proces Extract – Transform – Load

Używając Visual Studio utworzyć projekt typu Integration Services (wybierając z Menu File - > New Project) zawierający zapytania SQL opracowane w zadaniach 1-4.1 z listy 4 oraz w zadaniach 1-3 z listy 5.

Utworzony pakiet powinien działać sekwencyjnie i wykonywać następujące zadania:

- a) Usunąć tabele z przedrostkiem DIM i FACT (oczywiście usunąć tylko te, które istnieją)



b) Utworzyć table z przedrostkiem DIM i FACT

Execute T-SQL Statement Task

Connection: AdventureWorks New...

Execution time out: 0

T-SQL statement:

```
CREATE TABLE Czech.DIM_CUSTOMER (
  CustomerID INT NOT NULL,
  FirstName NVARCHAR(50) NOT NULL,
  LastName NVARCHAR(50) NOT NULL,
  Title NVARCHAR(10),
  City NVARCHAR(50),
  TerritoryName NVARCHAR(50),
  CountryRegionCode NVARCHAR(10),
  [Group] NVARCHAR(50),
);

CREATE TABLE Czech.DIM_PRODUCT (
  ProductID INT NOT NULL,
  Name NVARCHAR(100) NOT NULL,
```

OK Cancel View T-SQL Help

c) Wypełnić table danymi (instrukcje INSERT INTO)

Execute T-SQL Statement Task

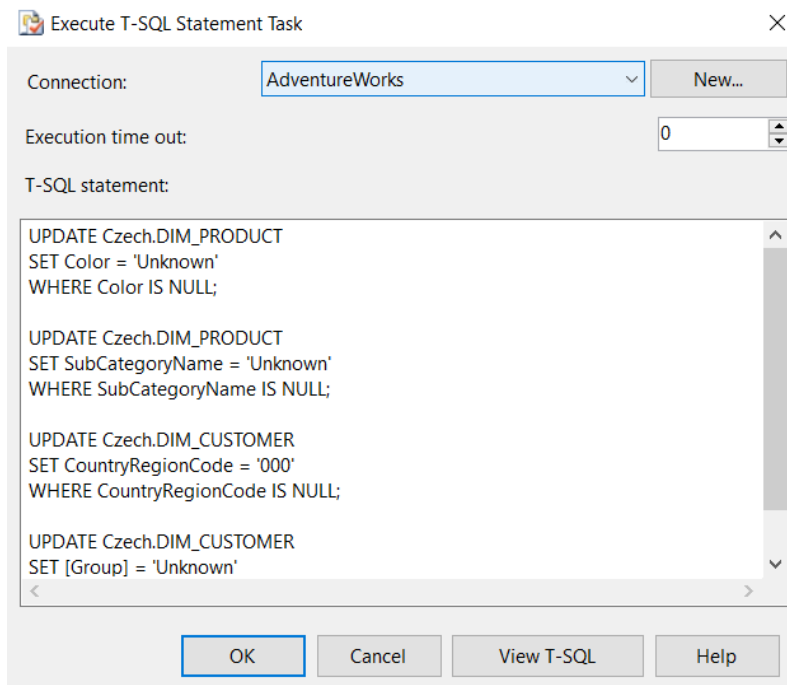
Connection: AdventureWorks New...

Execution time out: 0

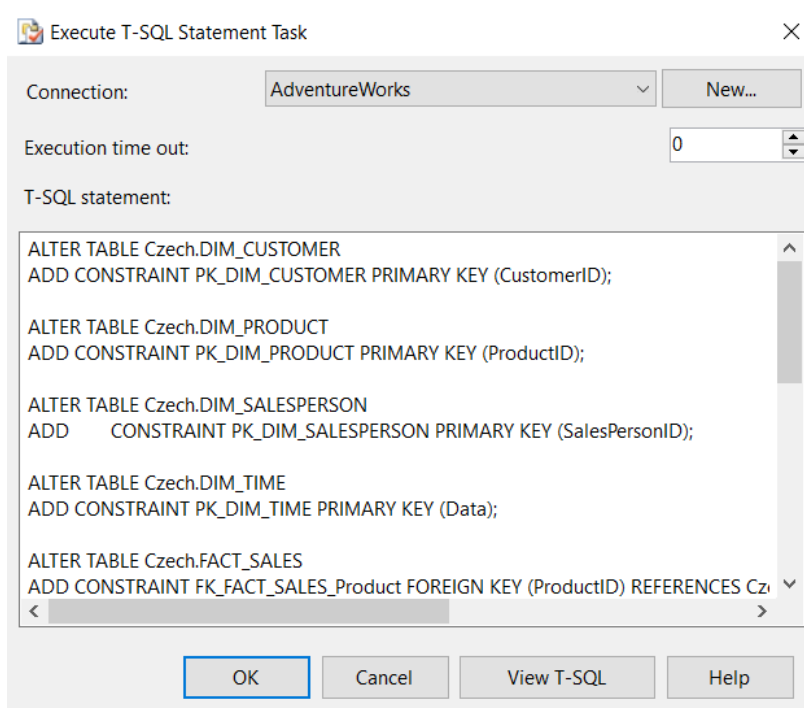
T-SQL statement:

```
INSERT INTO Czech.DIM_PRODUCT (ProductID, Name, ListPrice, Color, SubCategoryName)
SELECT DISTINCT
  p.ProductID,
  p.Name,
  p.ListPrice,
  p.Color,
  psc.Name AS SubCategoryName,
  pc.Name AS CategoryName,
  p.Weight,
  p.Size,
  1 AS IsPurchased
FROM Production.Product p
LEFT JOIN Production.ProductSubcategory psc ON p.ProductSubcategoryID = psc.ProductSubcategoryID
LEFT JOIN Production.ProductCategory pc ON psc.ProductCategoryID = pc.ProductCategoryID
```

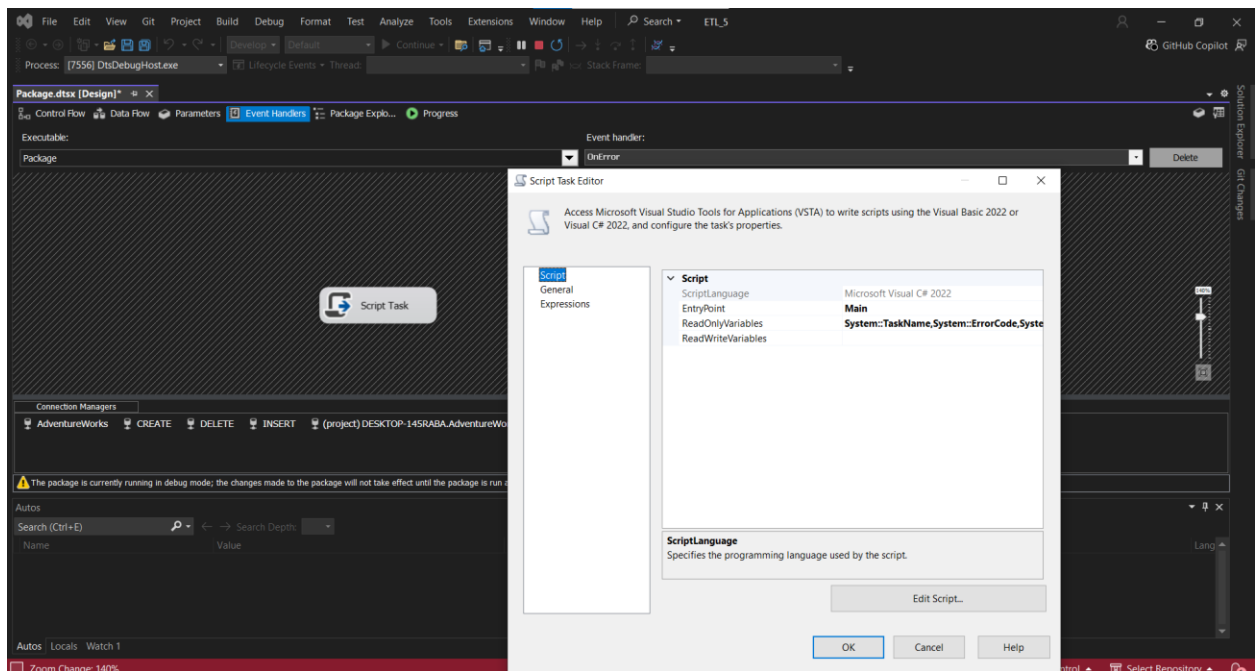
OK Cancel View T-SQL Help



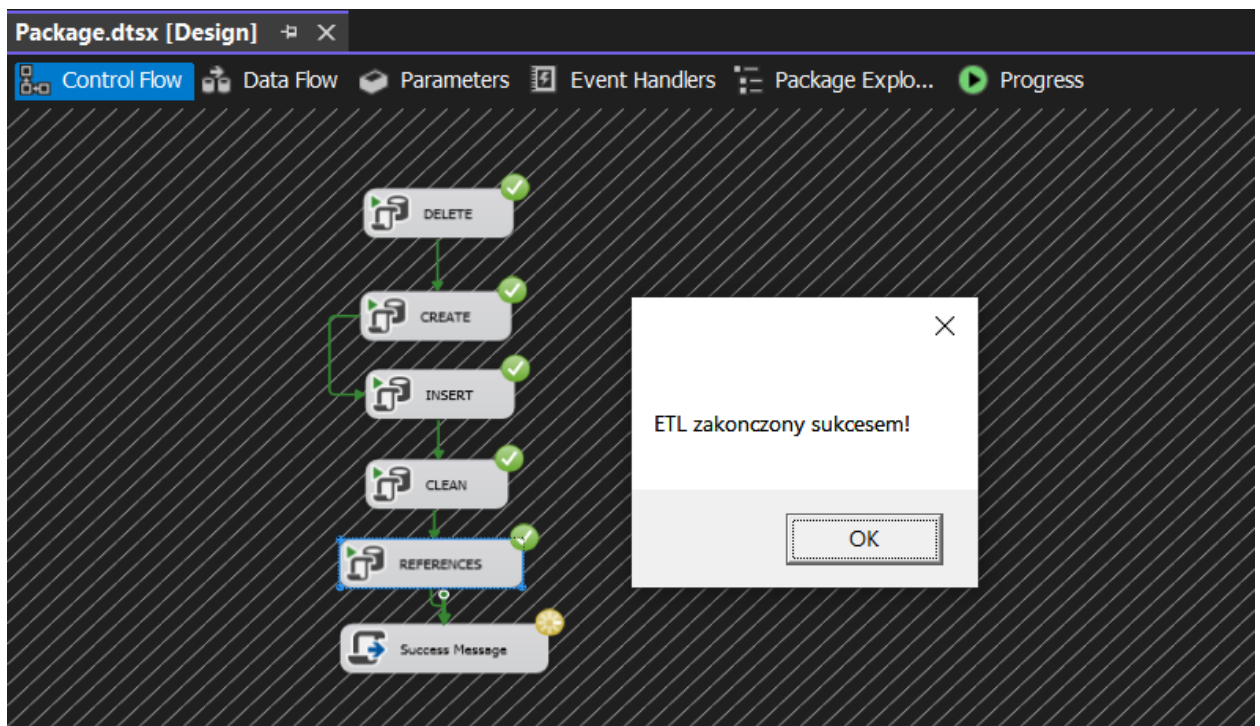
- d) Dodać więzy integralności z zadania 4.1 z listy 4 (bez sprawdzania poprawności integralności)













e) Obsłużyć błędy i wyjątki – zakładka Event Handlers



f) Wyświetlić informację o pozytywnie zakończonym procesie



Tabele zostały utworzone pomyślnie:

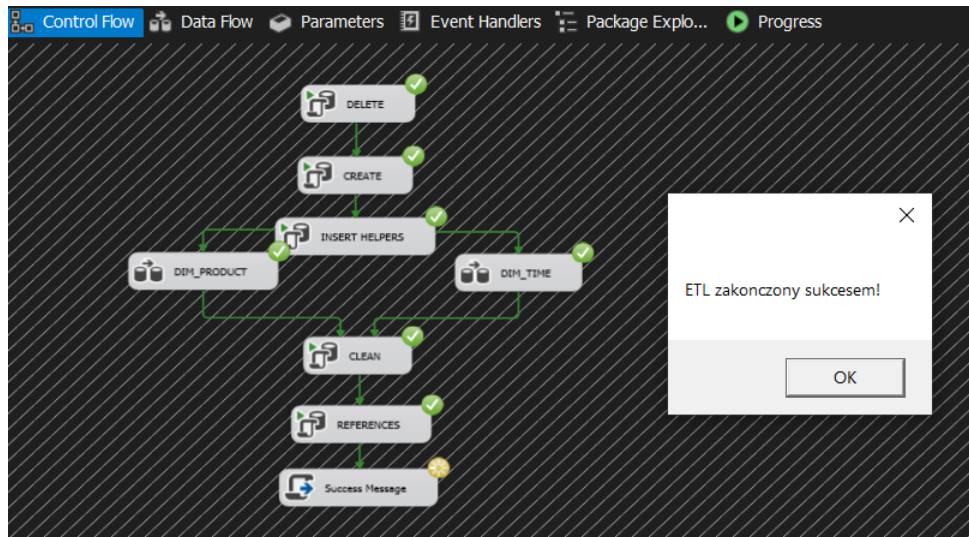
-   Czech.DIM_CUSTOMER
-   Czech.DIM_PRODUCT
-   Czech.DIM_SALESPERSON
-   Czech.DIM_TIME
-   Czech.FACT_SALES

Przykładowe dane z DIM_PRODUCT

	ProductID	Name	ListPrice	Color	SubCategoryName	CategoryName	Weight	Size	IsPurchased
90	807	HL Headset	124.73	Unknown	Headsets	Components	NULL	NULL	1
91	808	LL Mountain Handlebars	44.54	Unknown	Handlebars	Components	NULL	NULL	1
92	809	ML Mountain Handlebars	61.92	Unknown	Handlebars	Components	NULL	NULL	1
93	810	HL Mountain Handlebars	120.27	Unknown	Handlebars	Components	NULL	NULL	1
94	811	LL Road Handlebars	44.54	Unknown	Handlebars	Components	NULL	NULL	1
95	813	HL Road Handlebars	120.27	Unknown	Handlebars	Components	NULL	NULL	1
96	814	ML Mountain Frame - Black,...	348.76	Black	Mountain Frames	Components	2.73	38	1
97	815	LL Mountain Front Wheel	60.75	Black	Wheels	Components	NULL	NULL	1
98	816	ML Mountain Front Wheel	209.03	Black	Wheels	Components	NULL	NULL	1
99	817	HL Mountain Front Wheel	300.22	Black	Wheels	Components	NULL	NULL	1
100	819	ML Road Front Wheel	248.39	Black	Wheels	Components	850.00	NULL	1

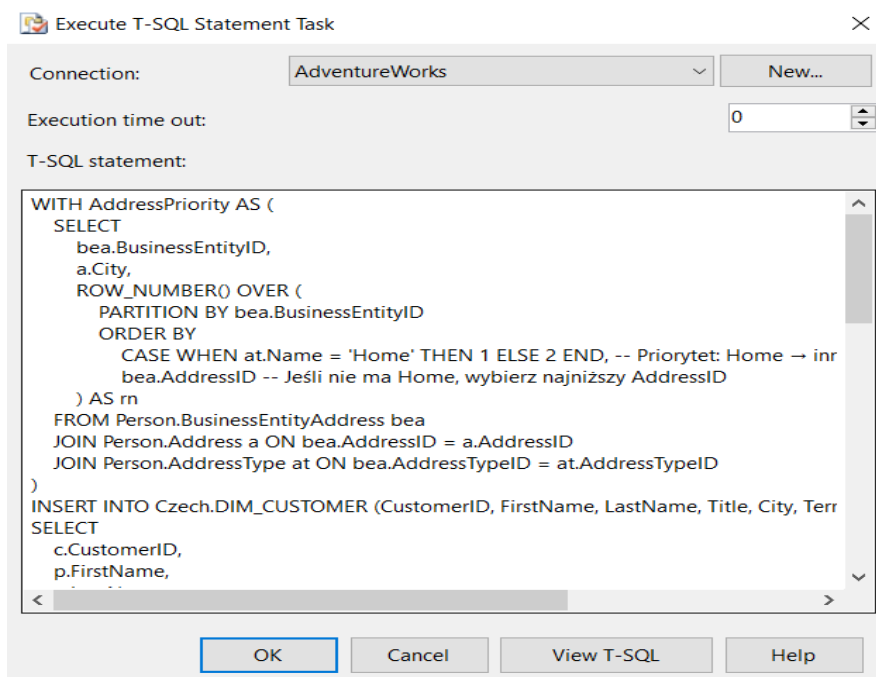
Zadanie 5. ETL (prawie) bez SQLa

Przygotować proces ETL analogiczny do opisanego w zad. 4. Dla wymiaru czasowego i co najmniej jednego innego wymiaru przygotować import danych korzystając z narzędzi dostępnych w zakładce Data Flow, m.in. OLE DB Source/Destination, Merge Join, Sort, Transformations, Derived Column, Fuzzy Lookup, Fuzzy Grouping, itp.



Insert Helpers:

W SQL Execution Task Insert Helpers zostawione zostały kwerendy uzupełniające dane o tabelach DIM_Customer, DIM_SalesPerson oraz FACT_Sales.



DIM_TIME:



Strategia wypełnienia DIM_TIME danymi:

- Wygenerowanie danych o wszystkich datach z zakresu zebranych danych (31.05.2011 – 07.07.2014) – skrypt C#
- Wypełnienie tabeli DIM_TIME wygenerowanymi danymi

Generacja danych:

```
2 references
public override void CreateNewOutputRows()
{
    DateTime startDate = new DateTime(2011, 5, 31);
    DateTime endDate = new DateTime(2014, 7, 7);

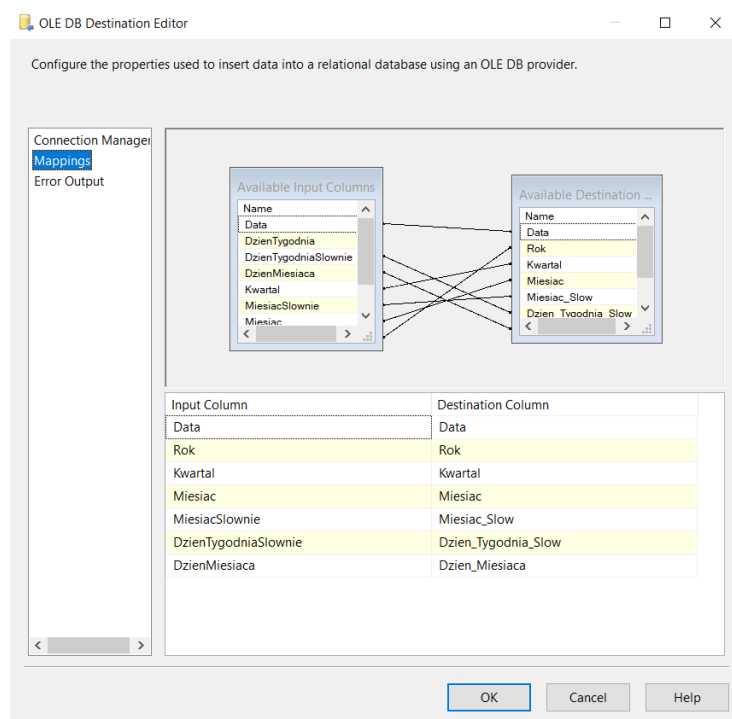
    while (startDate <= endDate)
    {
        Output0Buffer.AddRow();
        Output0Buffer.Data = (uint) int.Parse(startDate.ToString("yyyyMMdd"));
        Output0Buffer.Rok = (ushort) startDate.Year;
        Output0Buffer.Kwartal = (byte) ((startDate.Month - 1) / 3 + 1);
        Output0Buffer.Miesiac = (byte) startDate.Month;
        Output0Buffer.DzienMiesiaca = (byte) startDate.Day;
        Output0Buffer.MiesiacSlownie = startDate.ToString("MMMM", new System.Globalization.CultureInfo("pl-PL"));
        Output0Buffer.DzienTygodniaSlownie = startDate.ToString("dddd", new System.Globalization.CultureInfo("pl-PL"));
        startDate = startDate.AddDays(1);
    }
}
```

Typy danych:

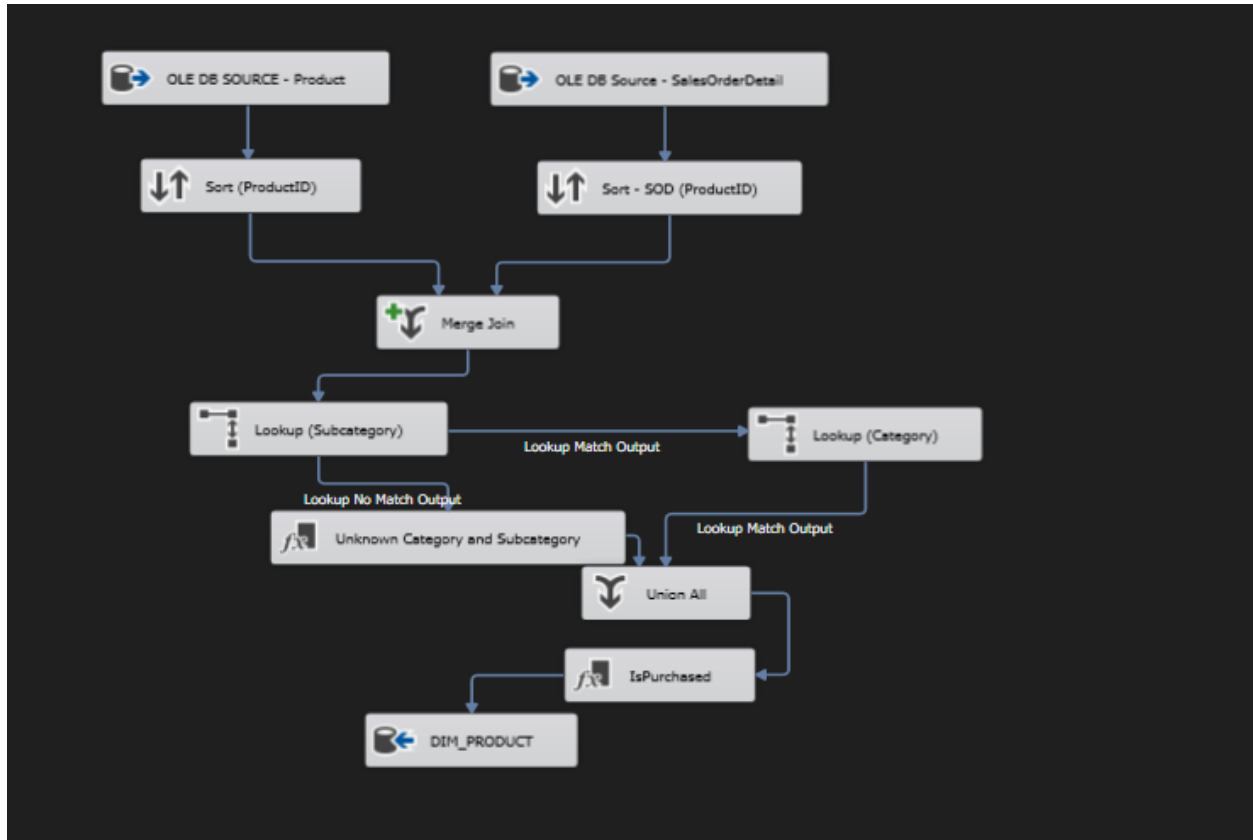
- Data – Unsigned 8-byte int
- Rok – Unsigned 2-byte int
- Kwartal – Unsigned 1-byte int
- Miesiac – Unsigned 1-byte int

- DzieńMiesiaca – Unsigned 1-byte int
- MiesiacSloownie – String
- DzieńTygodniaSloownie - string

Uzupełnienie tabeli DIM_TIME:



DIM_PRODUCT:



Strategia wypełnienia DIM_PRODUCT:

- Pobieramy interesujące nas dane z tabeli Products
- Pobieramy ProductID z SalesOrderDetail (wszystkie produkty które się sprzedały chociaż raz)
- Sortujemy dane z tabeli Product po ProductID
- Sortujemy dane z tabeli SalesOrderDetail po ProductID z użyciem opcji Remove rows with duplicate sort values, aby pozbyć się duplikatów
- Merge Join danych o Produktach z ProductID produktów, które się sprzedały, aby zostawić tylko informację o produktach, które zostały zakupione.
- Lookup z tabelą SubCategory:
 - Jeśli w produkcie istnieje SubCategoryId – Match – wysyłamy dane do Lookup Category – gdzie dodajemy informację o kategorii
 - Jeśli w produkcie nie istnieje SubCategoryId – No Match – wysyłamy dane do Derived Column (Unknown Category and SubCategory) – gdzie dodajemy pola CategoryName i SubcategoryName wypełnione wartościami „Unknown”

- Łączymy dane o kategoriach i wysyłamy je do DerivedColumn (IsPurchased) gdzie dodajemy informację o tym czy produkt jest kupiony (**zgodnie z instrukcją**)
- Uzupełniamy tabelę DIM_PRODUCT przygotowanymi danymi.

Wnioski:

- **Proces ETL może być w pełni zautomatyzowany** w ramach jednego pakietu SSIS, obejmującego usuwanie, tworzenie i uzupełnianie tabel danymi. Dzięki zastosowaniu Execute SQL Task i Data Flow Task, możliwe jest rozdzielenie logiki tworzenia struktur i ładowania danych.
- Możemy zoptymalizować tworzenie **Wymiaru Czasu (DIM_TIME)** za pomocą jednokrotnego przejścia przez wszystkie daty z zakresu, bez potrzeby sprawdzania ich występowania przy przeglądaniu każdego rekordu tabeli **FACT_SALES**.
- Zmiana wartości **NULL** na neutralne oznaczenia („Unknown”, „000”) pozwala ujednolicić dane i przygotować je do dalszej analizy.
- **Proces ETL może być wykonany przy minimalnym użyciu języka SQL**. Korzystając z narzędzi takich jak **Sort, Merge Join, Lookup, Derived Column** czy **Union All** możemy stworzyć bardzo zaawansowane zapytania wypełniające tabele danymi, bez użycia SQL.
- **Przy przetwarzaniu danych z niepełnym dopasowaniem (np. brak kategorii/subkategorii)** ważne jest uwzględnienie **No Match Output** oraz uzupełnienie brakujących kolumn, żeby uniknąć problemów ze spójnością danych.
- Za pomocą EventHandlers jesteśmy w stanie dokładnie przeanalizować błędy występujące zarówno w danych fragmentach procesu ETL jak i w całym procesie.