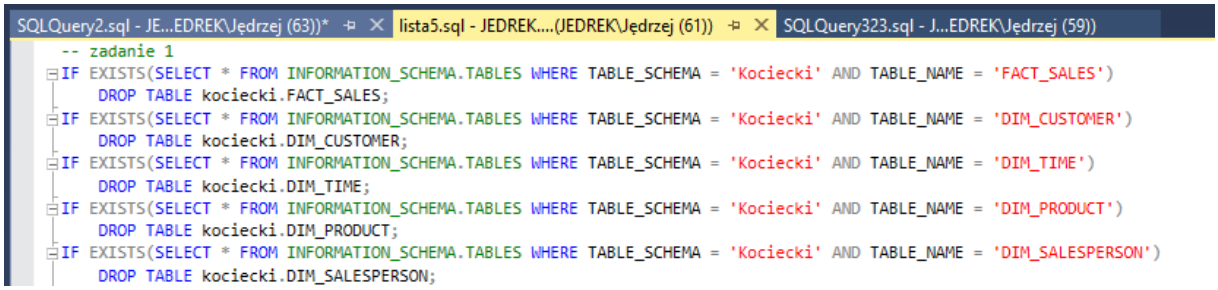
Zad. 1. Przygotowanie powtarzalności procesu ETL

Przygotować instrukcję usuwającą każdą z tabel utworzonych w trakcie pracy nad listą 4. Uwaga: Instrukcja powinna być wykonana tylko pod warunkiem istnienia usuwanej tabeli. Należy sprawdzić, czy dana tabela istnieje, używając instrukcji IF oraz informacji zawartych w widoku systemowym INFORMATION_SCHEMA.TABLES.



Zad. 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).

Tworzenie DIM_TIME

```
CREATE TABLE Kociecki.DIM_TIME
(
    PK_TIME INT PRIMARY KEY,
    Rok INT,
    Kwartal INT,
    Miesiac INT,
    Miesiac_slownie VARCHAR(20),
    Dzien_tyg_slownie VARCHAR(20),
    Dzein_miesiaca INT
);
```

Tworzenie tabeli z nazwami miesięcy

```
CREATE TABLE Kociecki.MONTHS_NAMES
(
    month_number INT,
    month_name VARCHAR(20)
);
```

Tworzenie tabeli z nazwami dni tygodnia

```sql
CREATE TABLE KOCIECKI.WEEKDAY_NAMES
(
    weekday_number INT,
    weekday_name VARCHAR(20)
);
```

Wypełnianie tabeli danymi

```sql
INSERT INTO kociecki.MONTHS_NAMES (month_number, month_name)
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');
```

```sql
INSERT INTO KOCIECKI.WEEKDAY_NAMES (weekday_number, weekday_name)
VALUES (1, 'poniedziałek'),
       (2, 'wtorek'),
       (3, 'środa'),
       (4, 'czwartek'),
       (5, 'piątek'),
       (6, 'sobota'),
       (7, 'niedziela');
```

```sql
USE AdventureWorks2019;
GO

WITH SourceDates AS (
    SELECT DISTINCT OrderDate AS CalendarDate
    FROM Sales.SalesOrderHeader
    WHERE OrderDate IS NOT NULL
    UNION
    SELECT DISTINCT ShipDate AS CalendarDate
    FROM Sales.SalesOrderHeader
    WHERE ShipDate IS NOT NULL
)
INSERT INTO Kociecki.DIM_TIME (
    PK_TIME,
    Rok,
    Kwartal,
    Miesiac,
    Miesiac_slownie,
    Dzien_tyg_slownie,
    Dzein_miesiaca
)
SELECT
    (DATEPART(year, sd.CalendarDate) * 10000) + (DATEPART(month, sd.CalendarDate) * 100) + DATEPART(day, sd.CalendarDate) AS PK_TIME,
    DATEPART(year, sd.CalendarDate) AS Rok,
    DATEPART(quarter, sd.CalendarDate) AS Kwartal,
    DATEPART(month, sd.CalendarDate) AS Miesiac,
    ISNULL(mn.month_name, 'Unknown') AS Miesiac_slownie,
    ISNULL(wn.weekday_name, 'Unknown') AS Dzien_tyg_slownie,
    DATEPART(day, sd.CalendarDate) AS Dzien_miesiaca
FROM
    SourceDates sd
    LEFT JOIN Kociecki.MONTHS_NAMES mn ON DATEPART(month, sd.CalendarDate) = mn.month_number
    LEFT JOIN Kociecki.WEEKDAY_NAMES wn ON DATEPART(weekday, sd.CalendarDate) = wn.weekday_number;
```

Więzy integralności (tak samo dla ShipDate)

```
ALTER TABLE KOCIECKI.FACT_SALES
ADD CONSTRAINT FK_TIME_ID
FOREIGN KEY (OrderDate) REFERENCES KOCIECKI.DIM_TIME(PK_TIME);
```

Przykładowy fragment danych, łącznie 1134 rekordy.

Results | Messages

	PK_TIME	Rok	Kwartal	Miesiac	Miesiac_slownie	Dzien_tyg_slownie	Dzein_miesiaca
1	20110531	2011	2	5	maj	2	31
2	20110601	2011	2	6	czerwiec	3	1
3	20110602	2011	2	6	czerwiec	4	2
4	20110603	2011	2	6	czerwiec	5	3
5	20110604	2011	2	6	czerwiec	6	4
6	20110605	2011	2	6	czerwiec	7	5
7	20110606	2011	2	6	czerwiec	1	6
8	20110607	2011	2	6	czerwiec	2	7
9	20110608	2011	2	6	czerwiec	3	8
10	20110609	2011	2	6	czerwiec	4	9
11	20110610	2011	2	6	czerwiec	5	10
12	20110611	2011	2	6	czerwiec	6	11
13	20110612	2011	2	6	czerwiec	7	12
14	20110613	2011	2	6	czerwiec	1	13
15	20110614	2011	2	6	czerwiec	2	14
16	20110615	2011	2	6	czerwiec	3	15
17	20110616	2011	2	6	czerwiec	4	16
18	20110617	2011	2	6	czerwiec	5	17
19	20110618	2011	2	6	czerwiec	6	18
20	20110619	2011	2	6	czerwiec	7	19
21	20110620	2011	2	6	czerwiec	1	20
22	20110621	2011	2	6	czerwiec	2	21
23	20110622	2011	2	6	czerwiec	3	22
24	20110623	2011	2	6	czerwiec	4	23
25	20110624	2011	2	6	czerwiec	5	24

Query executed successfully. | JEDREK (15.0 RTM) | JEDREK\Jędrzej (66) | AdventureWorks2019 | 00:00:00 | 1 134 rows

Zad. 3. Elementarne czyszczenie danych

Zamienić wszystkie wartości NULL

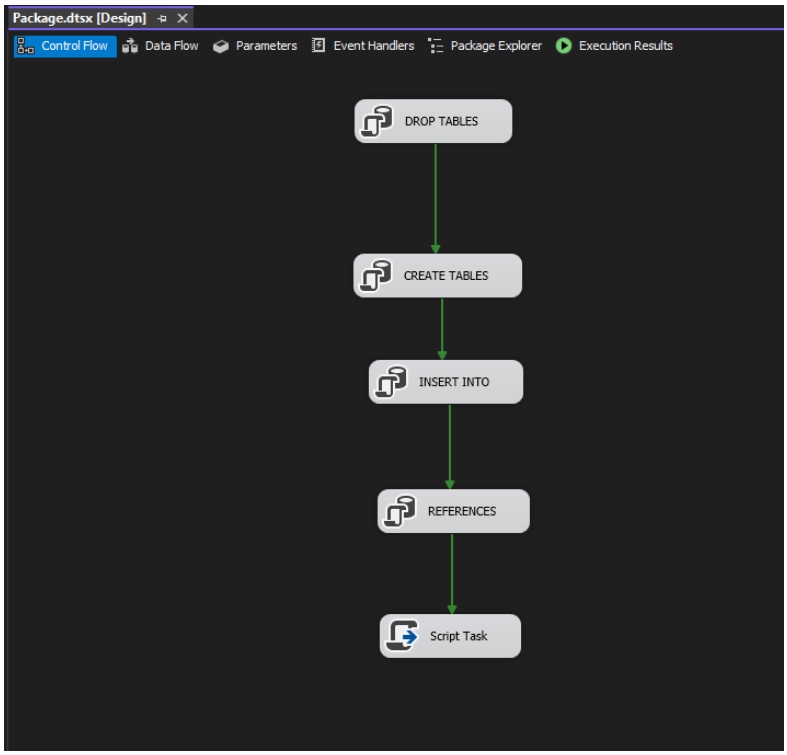
```sql
UPDATE kociecki.DIM_PRODUCT
SET Color = 'UNKNOWN'
WHERE Color IS NULL;

UPDATE kociecki.DIM_PRODUCT
SET SubCategoryName = 'UNKNOWN'
WHERE SubCategoryName IS NULL;

UPDATE kociecki.DIM_CUSTOMER
SET [Group] = 'UNKNOWN'
WHERE [Group] IS NULL;

UPDATE kociecki.DIM_SALESPERSON
SET [Group] = 'UNKNOWN'
WHERE [Group] IS NULL;

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

UPDATE kociecki.DIM_SALESPERSON
SET CountryRegionCode = '000'
WHERE CountryRegionCode IS NULL;
```

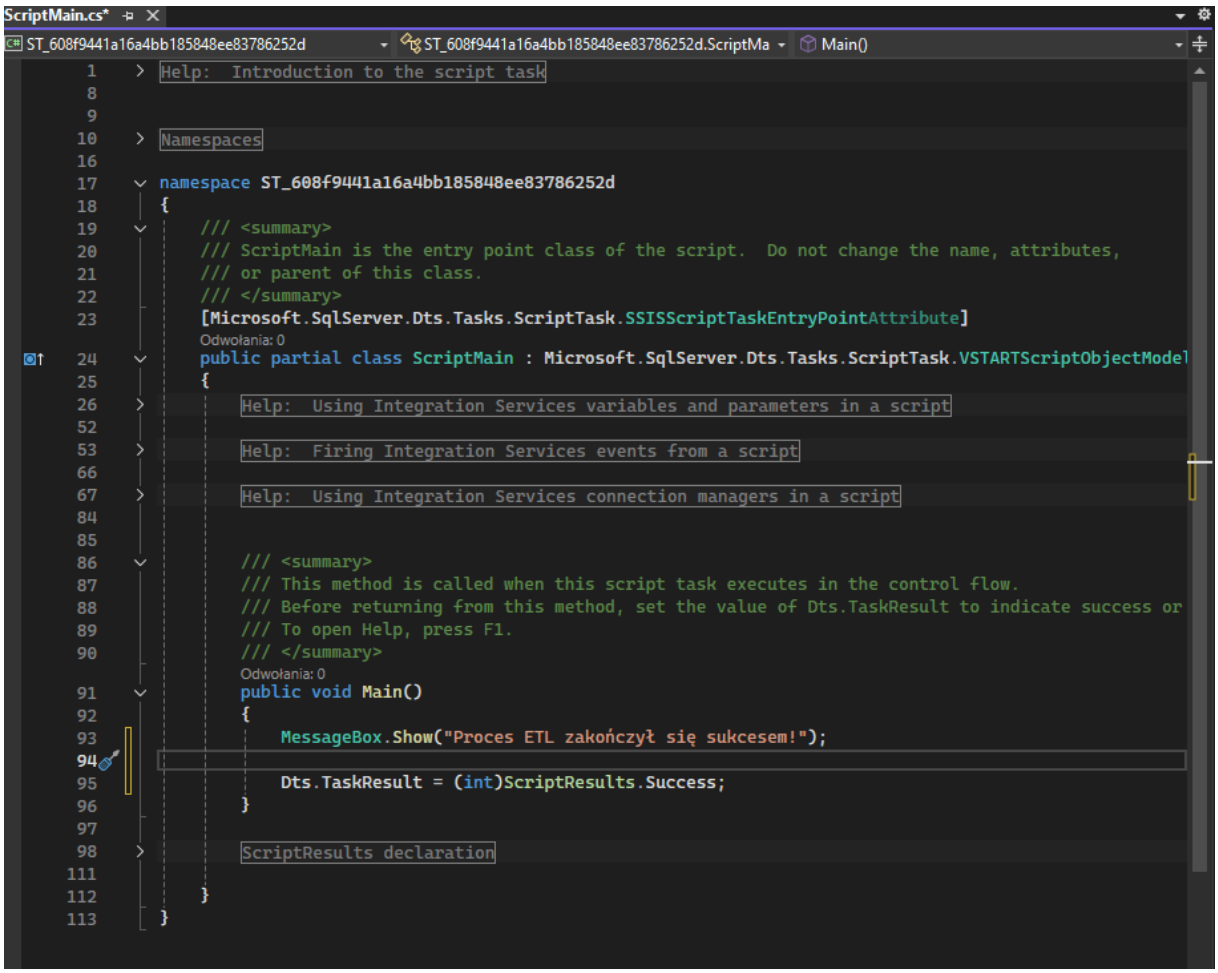
Zad. 4. Proces Extact – Transform - Load

Używając Visual Studio utworzyć projekt typu Integration Services (wybierając z Menu File - > New Project)

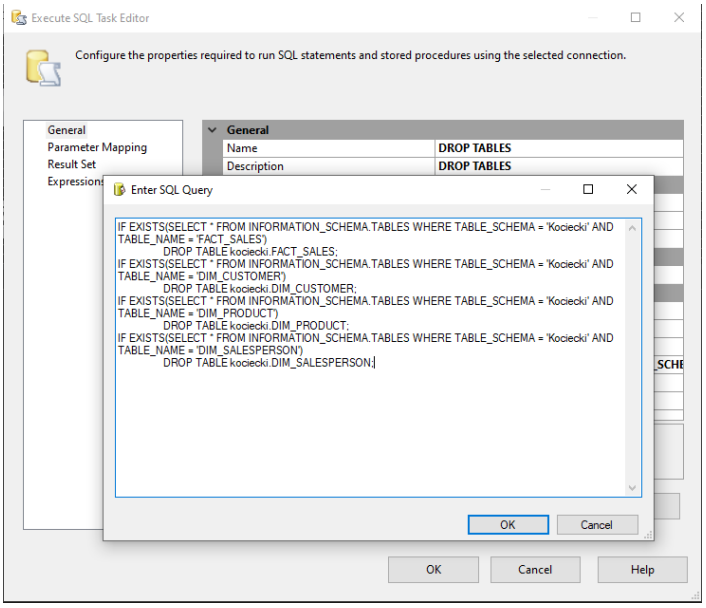
Na powstały proces ETL składa się:

- Usunięcie istniejących obiektów (jeżeli istnieją)
- Utworzenie tabel wymiarów i wypełnienie ich wyczyszczonymi danymi
- Wypełnienie tabeli faktów
- Wprowadzenie więzów integralności
- Obsługa błędow (Event Handlers)

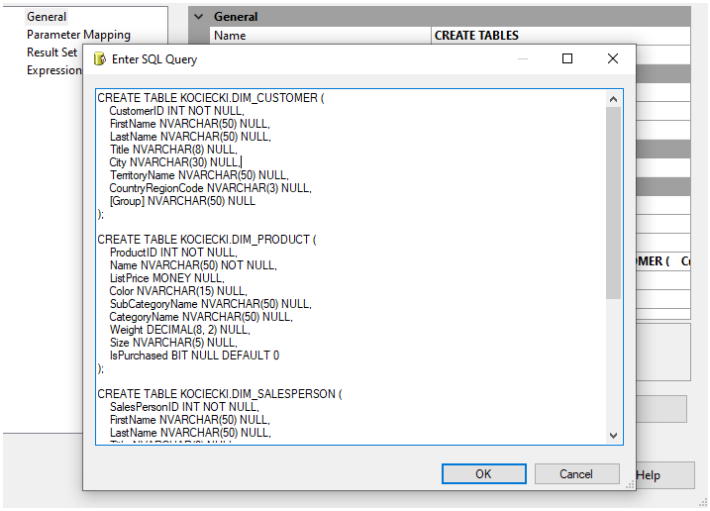

Package.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Execution Results
DROP TABLES
CREATE TABLES
INSERT INTO
REFERENCES
Script Task


ScriptMain.cs*
ST_608f9441a16a4bb185848ee83786252d
ST_608f9441a16a4bb185848ee83786252d.ScriptMa
Main()
Help: Introduction to the script task
Namespaces
namespace ST_608f9441a16a4bb185848ee83786252d
{
/// <summary>
/// ScriptMain is the entry point class of the script. Do not change the name, attributes,
/// or parent of this class.
/// </summary>
[Microsoft.SqlServer.Dts.Tasks.ScriptTask.SSISScriptTaskEntryPointAttribute]
Odwołania: 0
public partial class ScriptMain : Microsoft.SqlServer.Dts.Tasks.ScriptTask.VSTARTScriptObjectModel
{
Help: Using Integration Services variables and parameters in a script
Help: Firing Integration Services events from a script
Help: Using Integration Services connection managers in a script
/// <summary>
/// This method is called when this script task executes in the control flow.
/// Before returning from this method, set the value of Dts.TaskResult to indicate success or
/// To open Help, press F1.
/// </summary>
Odwołania: 0
public void Main()
{
MessageBox.Show("Proces ETL zakończył się sukcesem!");
Dts.TaskResult = (int)ScriptResults.Success;
}
ScriptResults declaration
}
}

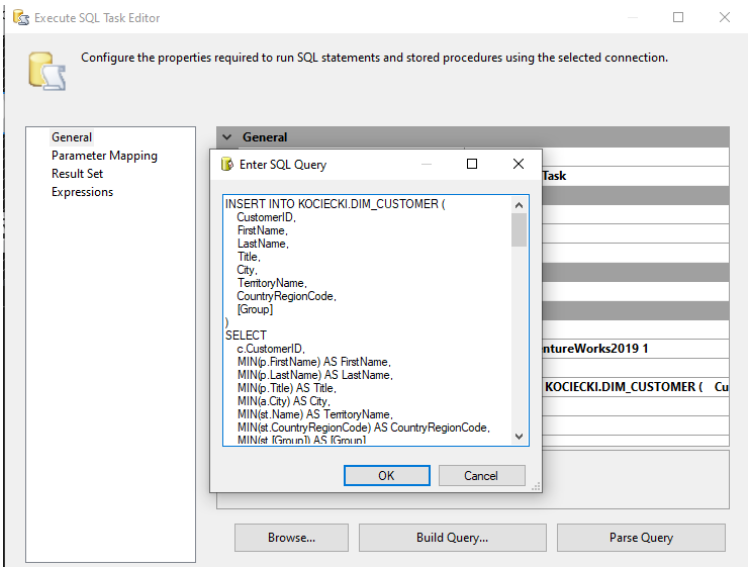
a) Usunąć tabele z przedrostkiem DIM i FACT (oczywiście usunąć tylko te, które istnieją),



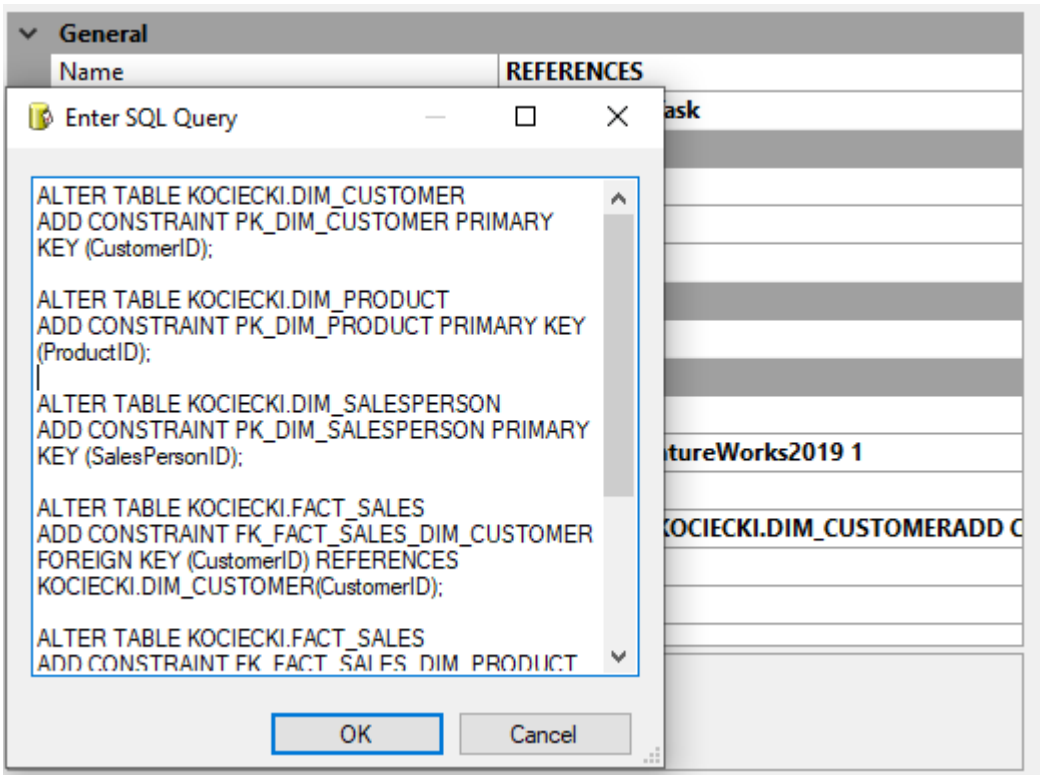
b) Utworzyć tabele z przedrostkiem DIM i FACT,



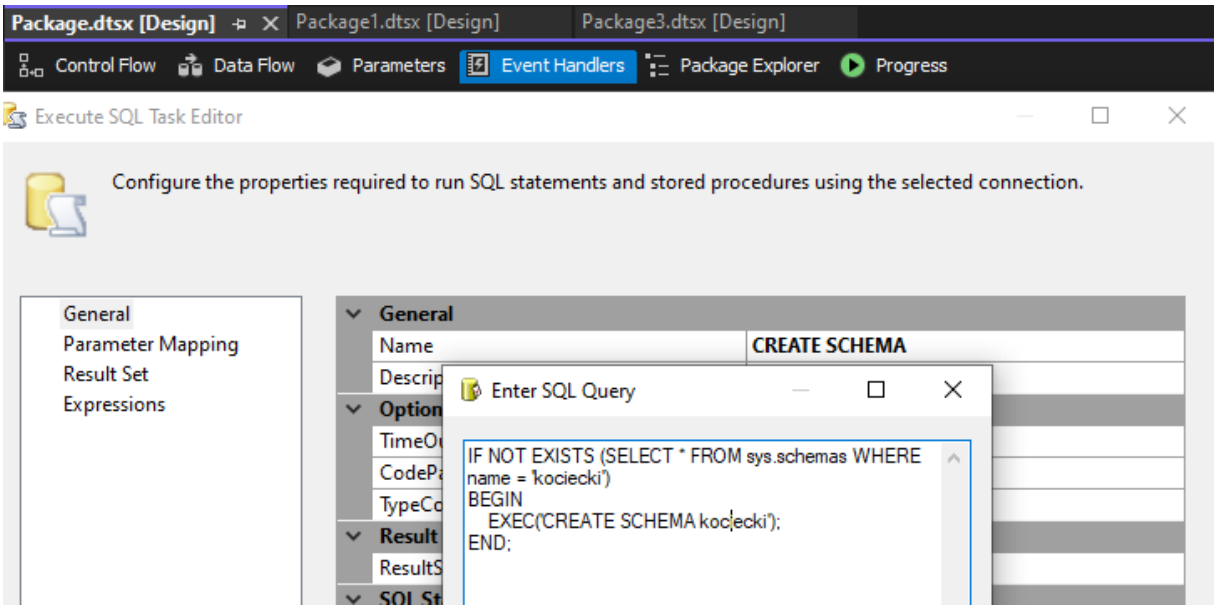
c) Wypełnić tabele danymi (instrukcje INSERT INTO),
Również podczas Insertowania, zostało zadbane ujednolicanie wartości niektórych wybrakowanych zmiennych.



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



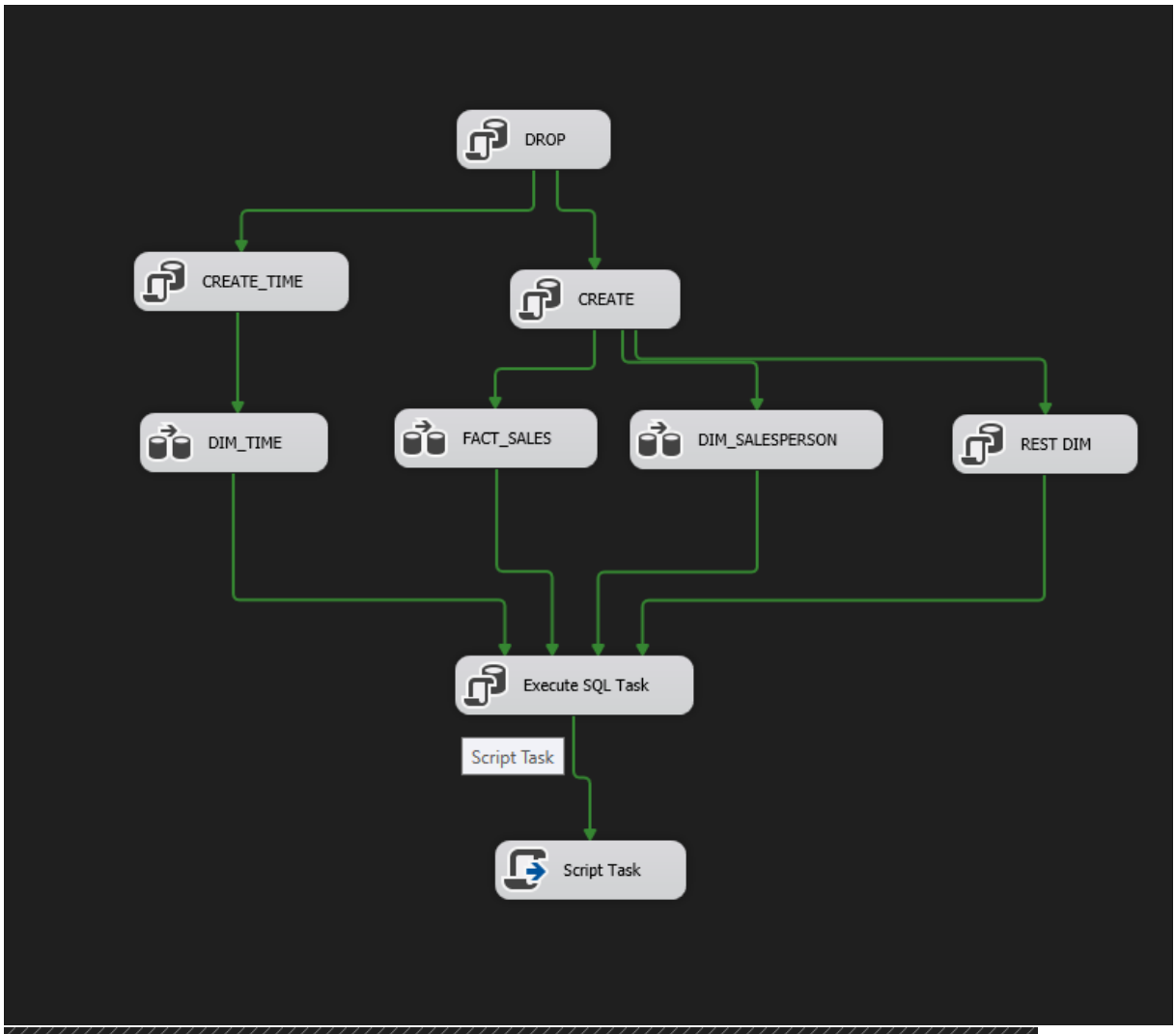
e) Obsłużyć błędy i wyjątki – zakładka Event Handlers, f) Wyświetlić informację o pozytywnie zakończonym procesie.

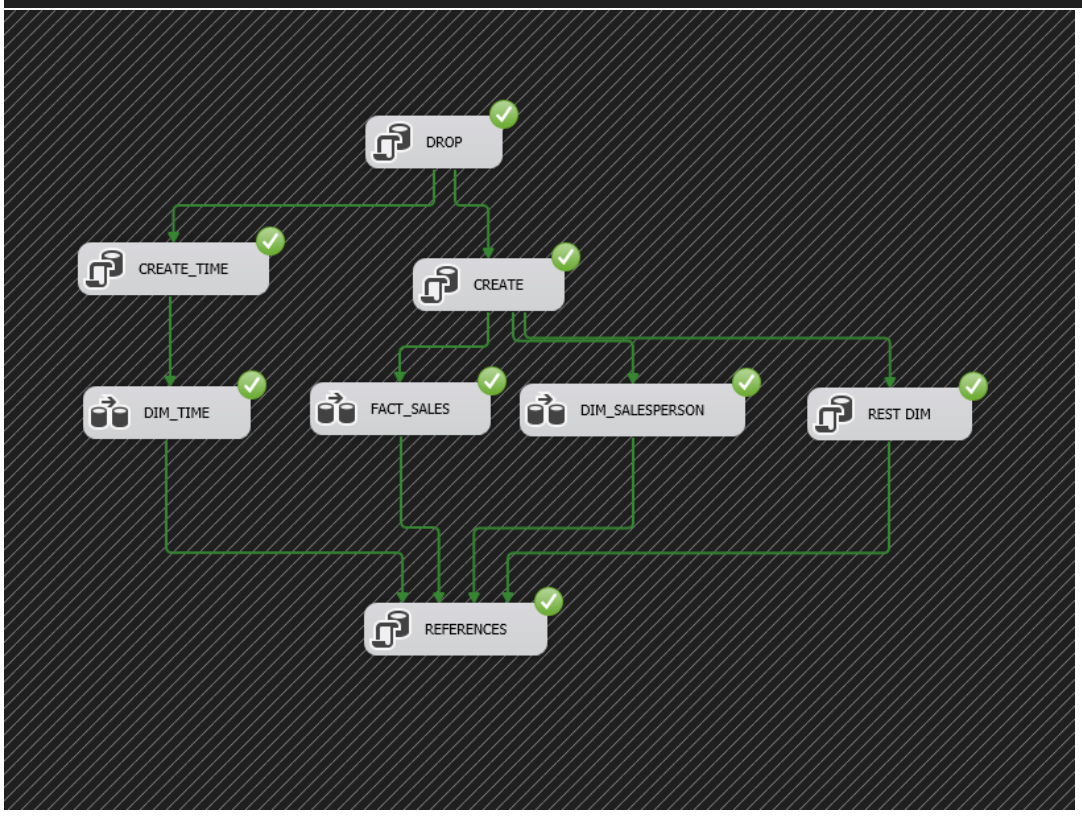


Zad. 5. ETL (prawie) bez SQLa

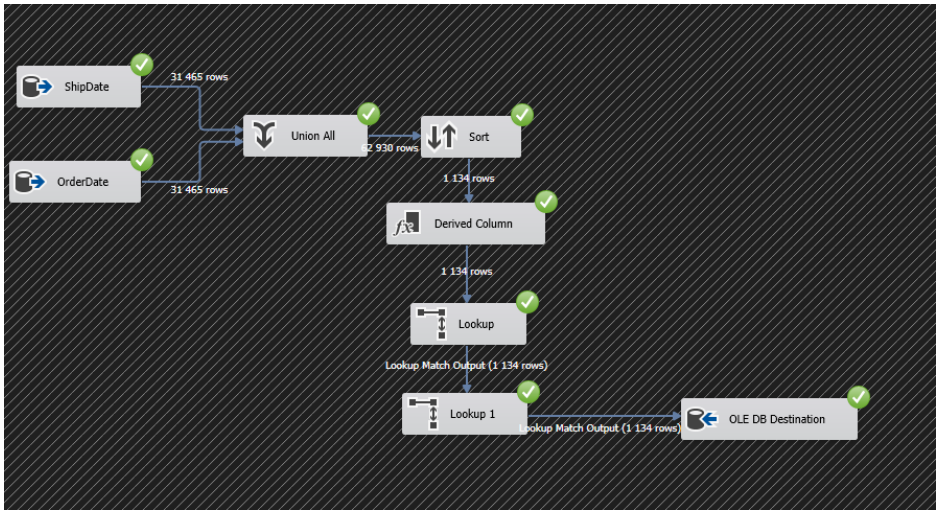
Przygotować proces ETL analogiczny do opisanego w zad. 4. Dla wymiaru czasowego i co najmniej jednego innego wymiaru przygotować import danych korzystając z narzędzi dostępnych w zakładce Data Flow

Control Flow pakieru SSIS, zmodyfikowany w celu użycia Data Flow Tasków dla wybranych wymiarów

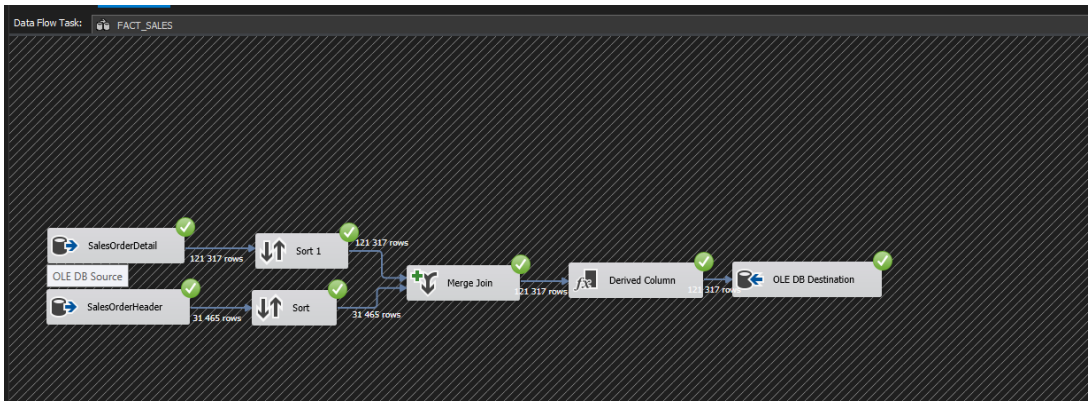

DROP
CREATE_TIME
CREATE
DIM_TIME
FACT_SALES
DIM_SALESPERSON
REST DIM
Execute SQL Task
Script Task
Script Task


DROP
CREATE_TIME
CREATE
DIM_TIME
FACT_SALES
DIM_SALESPERSON
REST DIM
REFERENCES

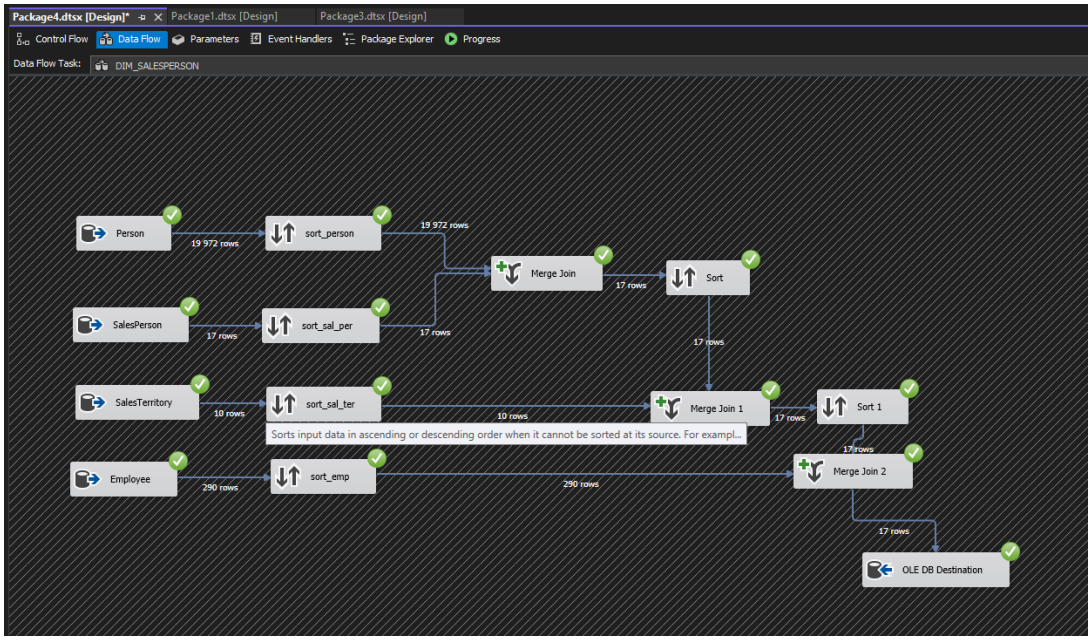
Data Flow dla DIM_TIME



Data Flow dla FACT_SALES



Data Flow dla DIM_SALESPERSON



Wnioski:

Ostatnie zadanie polegało na modyfikacji zadania 4, tak aby proces ETL korzystał również z komponentów *Data Flow.* Utworzone zostały 3 takie komponenty, odpowiedzialne za wypełnianie tabel danymi. Wykorzystano w tym celu między innymi:

OLE DB Source – które służy do pobierania danych ze źrodłowych tabel

Derived Column, które służy do utworzenia nowych kolumn lub zastąpienia istniejących, umożliwia transformacje istniejących danych poprzez wyrażenia i funkcje

Sort – wymagane przed łączeniem kolumn, umożliwiające również usuwanie duplikatów

Union All – łączenie pionowe danych

Lookup – dołączanie informacji z innych tabel

Fuzzy Lookup/*Grouping* – nieużywane, ale umożliwiające np. ujednolicanie nazw produktów

OLE DB Destination: Do zapisywania przetworzonych danych do docelowych tabel

Proces ETL zapewnia wysoką powtarzalność i stabilność całego procesu – w przypadku napotkania problemów wystarczy usunąć istniejące tabele i ponownie uruchomić pakiet, co znacząco usprawnia zarówno rozwój, jak i testowanie. Graficzne zadania typu Data Flow w SSIS umożliwiają przejrzyste śledzenie przepływu danych, szczególnie dla użytkowników nietechnicznych, jednak ich konfiguracja wymaga ręcznego przeklikiwania, co bywa bardziej czasochłonne niż napisanie analogicznego skryptu SQL.

Utworzenie dedykowanego wymiaru czasu z jednolitym kluczem (np. w formacie RRRRMMDD) i przypisanymi atrybutami, takimi jak rok, kwartał, miesiąc czy dzień tygodnia, stanowi fundament wszelkich analiz. Dzięki temu można łatwo agregować, filtrować i grupować dane w kontekście czasowym, co jest niezbędne przy budowie rzetelnych raportów.

Ostateczny wybór między podejściem wizualnym a bazującym na SQL powinien uwzględniać skalę projektu oraz kompetencje zespołu. Data Flow w SSIS zapewnia czytelność rozbudowanych procesów kosztem większego nakładu pracy konfiguracyjnej, natomiast skrypty SQL mogą być szybsze do przygotowania, lecz w miarę wzrostu złożoności trudniejsze w utrzymaniu.