Emilija Dimikj

May 10, 2020

BIDD 220 B Sp 20: Data Migration Techniques (ETL Processing)

Assignment04

SSIS Project for Flush & Fill Data Warehouse

This SSIS project contains two packages that perform same ETL process but in two different ways. Source is ***AdventureWorks\_Basics*** database and destination is ***DWAdventureWorks\_Basics*** data warehouse. ETL process is based on SQL script created in Assignment 1 of this course. Purpose of this assignment is to explore two different ways of developing SSIS Packages for the same ETL Process and to compare them.

In both packages, ETL process is done in 4 stages:

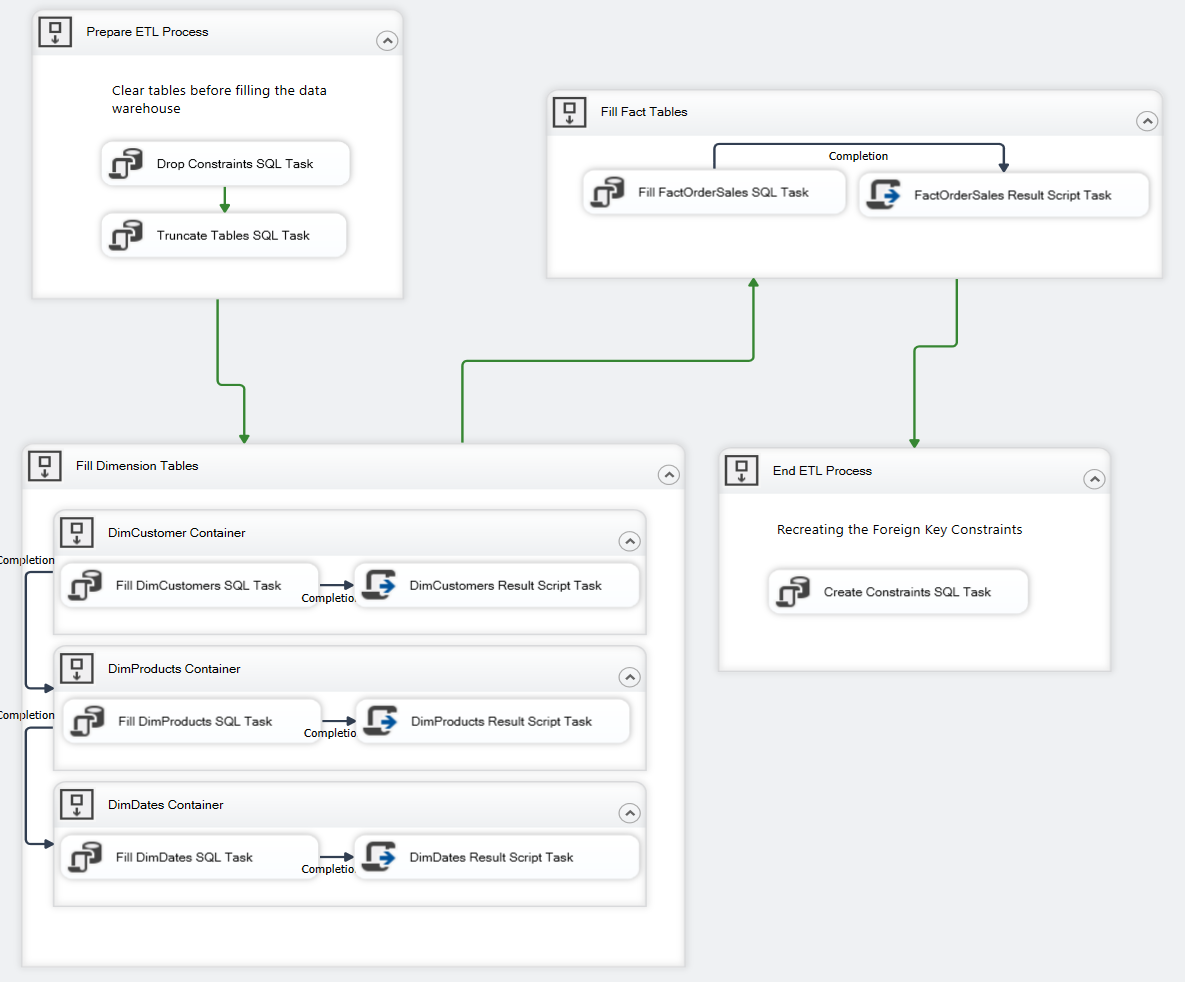
1. Prepare the data warehouse for the ETL process: dropping all constrains between tables and truncate all the tables
2. Fill Dimension tables: populating the dimension tables with data one by one. They are independent tables so there is no specific order of data population
3. Fill Fact table: populating the fact table with data
4. Ending the ETL process: creating all constrains between tables

# Package that uses SQL code - AdventureWorksETLWithSQLCode.dtsx

This package uses stored procedures created for ETL SQL script *(Figure 1)*. Those procedures are called with *Execute SQL Tasks*. This package uses just destination ADO.NET connection. All connectons fror extracting data are in the SQL stored procedures.

Each stored procedure has returning value to provide the stored procedure’s execution status to the SSIS *Execute SQL Task* (0 not executed, 1 executed with success, -1 error in execution process). Return code is read by *Script Task* and logged in log file ***ExecProc.log***. In order to reuse the same variable for each Execute SQL Task as return code, each SQL Task & Script Task are in own Sequence Container.

Precedence Constraints between Filling dimension table’s containers are set up upon completion status because all those tasks are independent from each other.

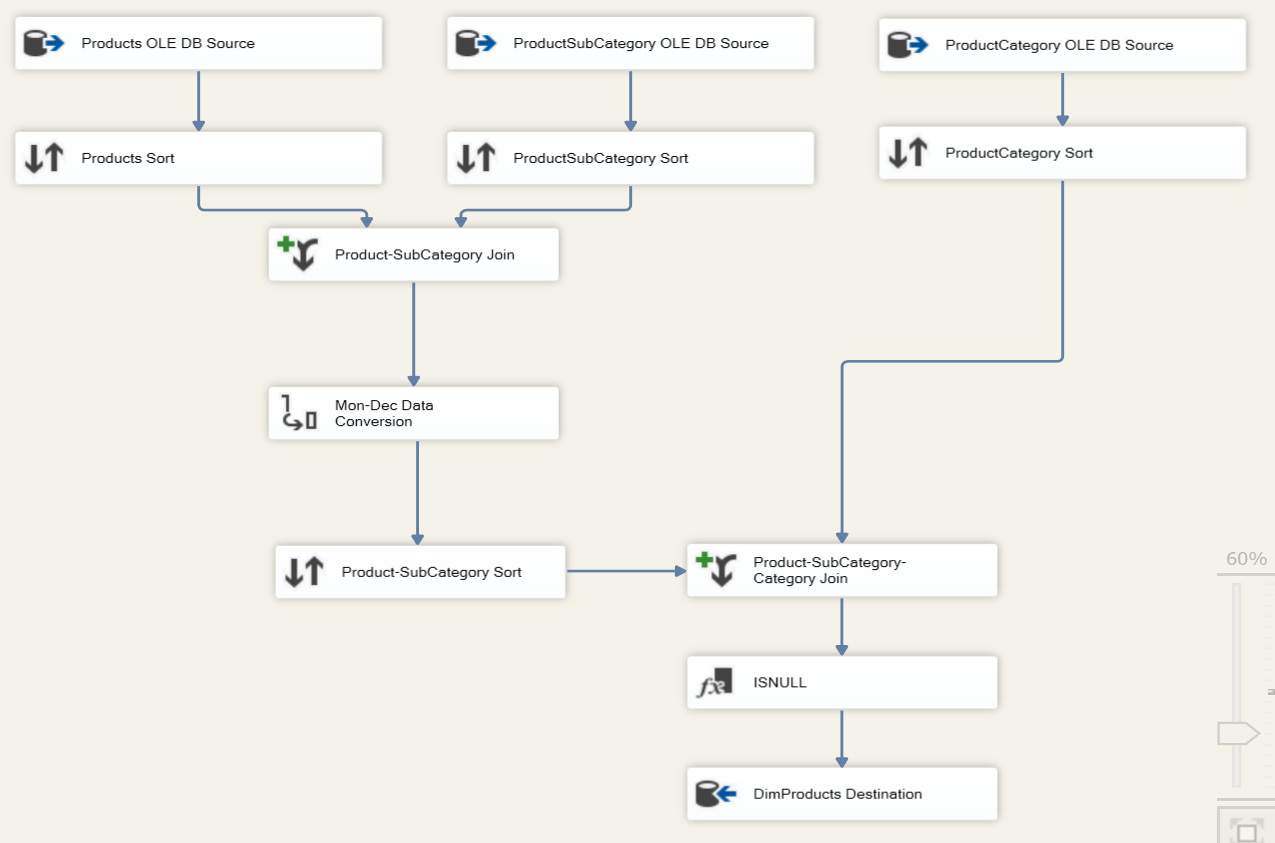


*Figure 1: AdventureWorksETLWithSQLCode package*

# Package that does not contain SQL code - AdventureWorksETLWithSSISTransformations.dtsx

This package pulls the data directly from the source database and uses SSIS Transformations and destinations to fill the data warehouse. In this package all transformations, joins, loops are done with Data Flow Tasks. SQL code for dropping and creating constraints and code for truncate tables is directly written as SQL statement in Execute SQL Task.

Each Data Flow task encapsulates the data flow engine that moves data between source and destination, and lets the user transform, clean, and modify data as it is moved.



*Figure 2: DimProduct Data Flow Task (joining 3 tables)*

# Comparation between the packages

|  |  |
| --- | --- |
| Hybrid model with using SQL | Clean SSIS without SQL |
| Works faster: everything is processed within the SQL engine. JOIN and MERGE statements works faster when dealing with a lot of records. | Works slower: All data is transferred over to the SSIS memory space and doing the manipulation there. LOOKUP tasks and SCD large tasks do not have good performance in case of big amount of records. |
| Current skill set:   * SQL: requires good knowledge of SQL, creating stored procedures, complex join statements and conversions. * SSIS: basic knowledge of SSIS | Current skill set:   * SQL: does not require extensive knowledge of SQL. * SSIS: good knowledge of SSIS, formula language |
| Less complex: in case of JOIN statements require less work. | More complex: in case of JOIN statements, one data flow can require creation of many tasks. |
| Easy to change: Majority of the code is in SQL stored procedures, scripts and functionality changes are done in one place. | Difficult to change: finding the right task, data flow, conversion … can be challenging and time consuming. |

# Summary

In this assignment I found that hybrid model works better and faster. Logging, auditing and error handling in this case was little bit more complicated to implement but once done is very versatile. Also, using graphic interface makes it easier to build large, complex and reliable data flows and is easier to visualize and to control. Using the hybrid model, allows us to make a change in the stored procedure, avoiding having to make a change in SSIS and re-deploying the package. The negative side of this is maintenance, instead of having everything contained within SSIS, we have code split between SSIS and SSMS and requires a balance and certain skill set.

In my opinion, this model is best of both worlds