Lab Summary

In this assessment you will revisit the package that you created in last lab for transforming and loading the Fact.EnrollmentOutcomes data. In the last lab, most of us did a large portion of transforming the staging data into the Fact table design via a custom SQL statement in the OLEDB Source. This works fine; however, from a visual documentation and debugging perspective, it hid many steps in the transformation process.

In this lab, you will research and discover how you can use different techniques to break the transformations up into a few components and steps.

The end goal for this assessment is for you to add a few more SSIS skills to your repertoire so you have more choices for development in the future!

Lab Files

Please download and run Lab6_Database.sql. It will create the same database that we used last assessment. Next, download the SSIS starter project from the Lab 6 link. Update the project level connection manager to use your database and credentials.

The lab files have a couple of stubbed practice packages for you to work on to test out new ideas. They each segregate getting a converted grade from getting a date key. The Final package - Integration-Transform-Load-EnrollmentOutcomes.dtsx – should be completed last, after you have worked on the two practice concepts.

Test-Package-DateKey: Practice Getting the Date Key

Currently this package shows getting the date key via a lookup component. This is another way to access the value based on term, day, and time fields from the staged section data. Review this componenent, and then delete it.

Part of the challenge in last lab was that you were given a scalar function to lookup the date key value, but the problem is that you cannot call a user-defined function in the data flow process. However, you can call a user defined stored procedure using an OLEDB Command Component! You have license to do one of 2 things: you can rewrite the scalar function so that it is a stored procedure instead, or you can create a stored procedure that calls this function. Your choice! There is a trick to executing a stored procedure in an OLEDB command component - part of it has to with mapping the resulting output parameter. For help on the set-up, check out this post

Modify the data flow tab in Test-Package-DateKey by adding in components between the lookup staging section component and the row count so you can acquire the date key using a stored procedure. Add a data viewer to the data path before the row count to view the results of the transformation when you run your project.

Test-Package-Convert-Grades: Practice Converting Grades via a Cache Connection

You have performed lookups on relational tables. Another way to dynamically access lookup values is via a Cache Transformation. Your starter file contains a .csv file with all of the grades converted to numeric values. Add a Cache Transformation between the Source and Row Count Components and configure it. You can choose to implement a file-based cache (.caw file) or in memory. Your choice. Add a data viewer to the data path before the row count to view the results of the transformation when you run your project.

NOTE: there is a hidden trick to getting this package to work! You should get an error message when you try to run the component. What is the problem and how can you fix it?

Finalizing the “Real” Transform-Load Package!

Now that you have lots of practice in on new skills, rework Final-Package-Load-Enrollment so that it uses your new, stepped, way of transforming staging data to the fact table to acquire the date key and grade conversion in one package. Test the package’s execution to confirm success. We won’t track lineage for this learning lab, so you can default the audit key to 0 for this practice.

Grading

Please show your working packages before you leave today’s class.