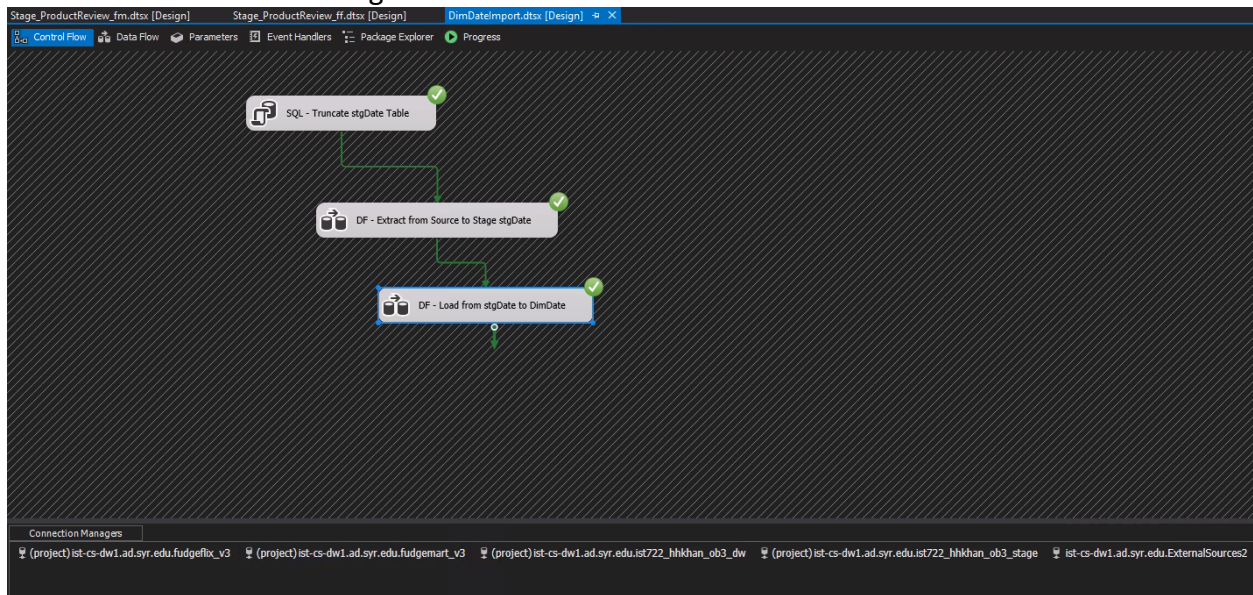


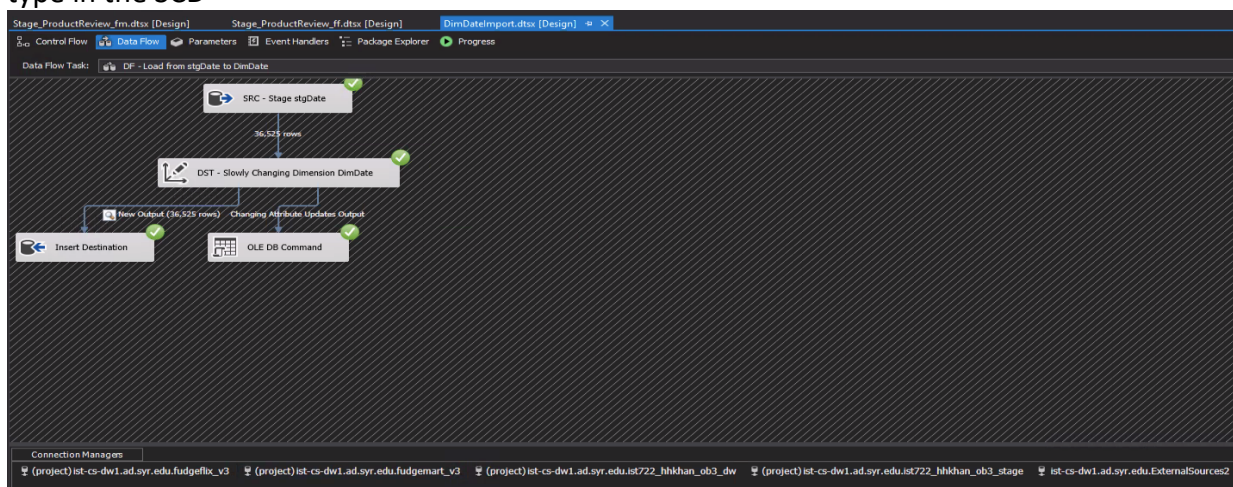
IST 722 Group 3

ETL – Documentation

For DimDateImport: Used Execute SQL Task where we connected to our stage and added a truncate table statement for stgDate table. For extracting data from source to destination we used DF-Extract from Source to Stage StgDate as seen below to take the data from External sources and load it to our stgDate table

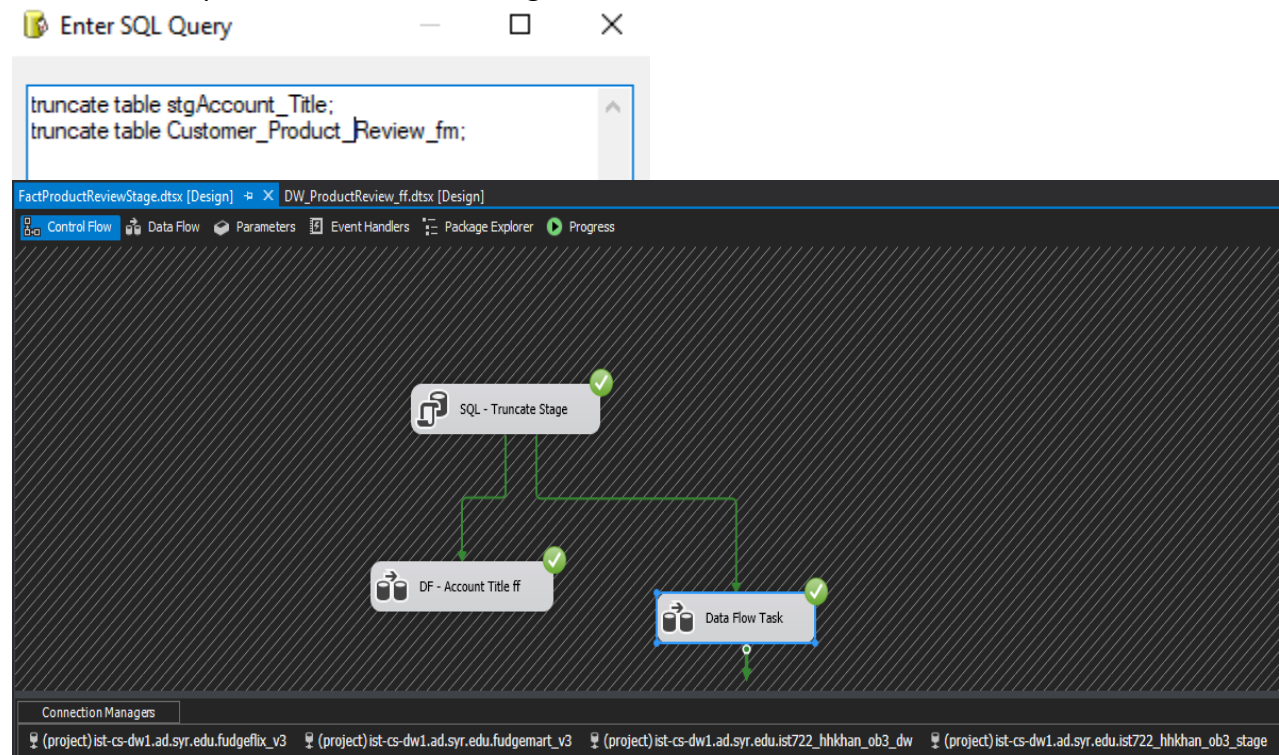


DimDateDataImport Flow: In the data flow below, we connected with our stage stgDate as a source and then by using SCD we connected to our data warehouse to pull data from source and push it to our data Warehouse DimDate table. We used Changing attribute as a change type in the SCD

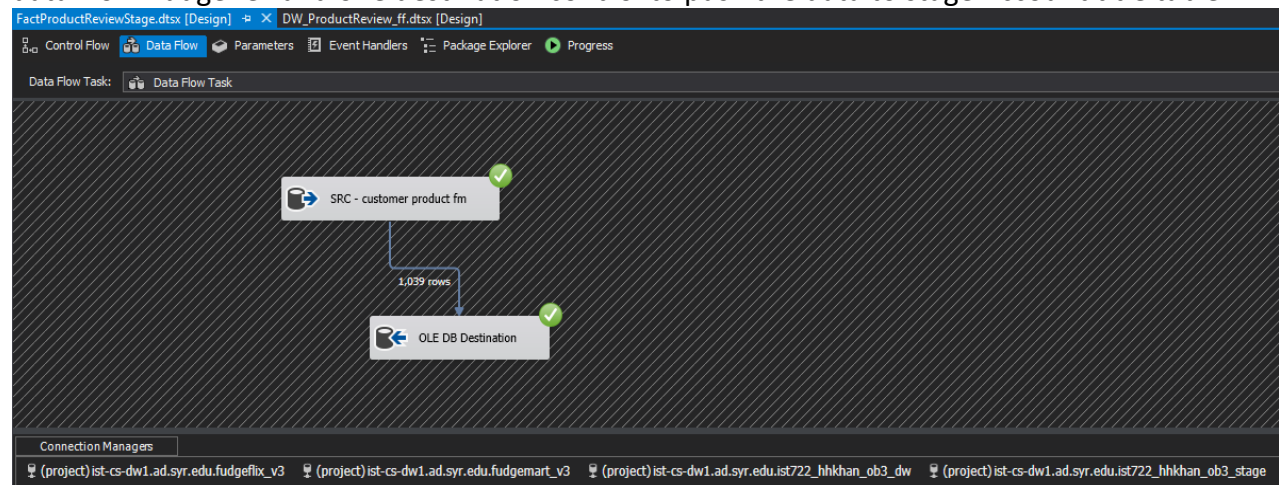


Fact Product Review Control Flow: We have used one Execute SQL Task and two dataflows for the to stage the FactProductReview table. One data flow for Account Title from Fudgefix and

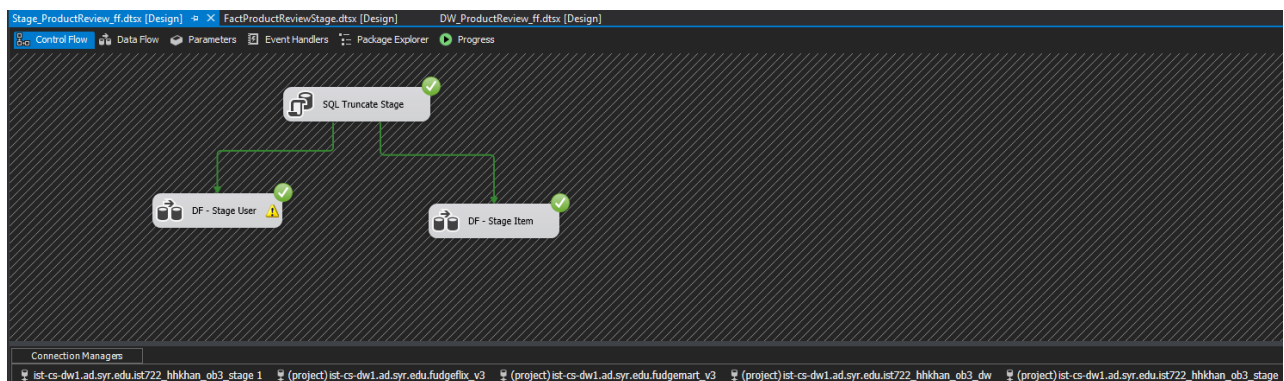
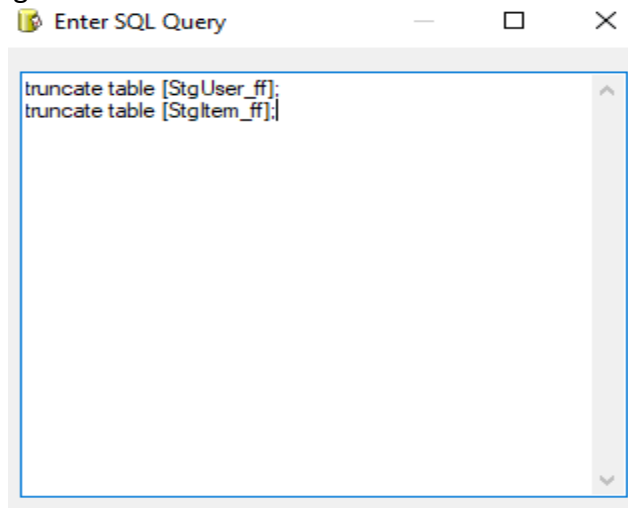
one Customer product review from Fudgemart where we truncated the two flows as follows



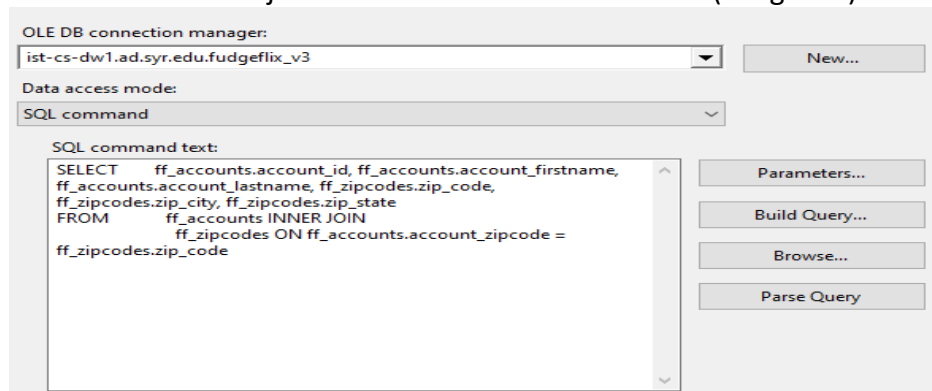
Fact Product Review Data Flow: The data flow shows how we used one source control to pull data from Fudgeflex and one destination control to push the data to stage Account title table.



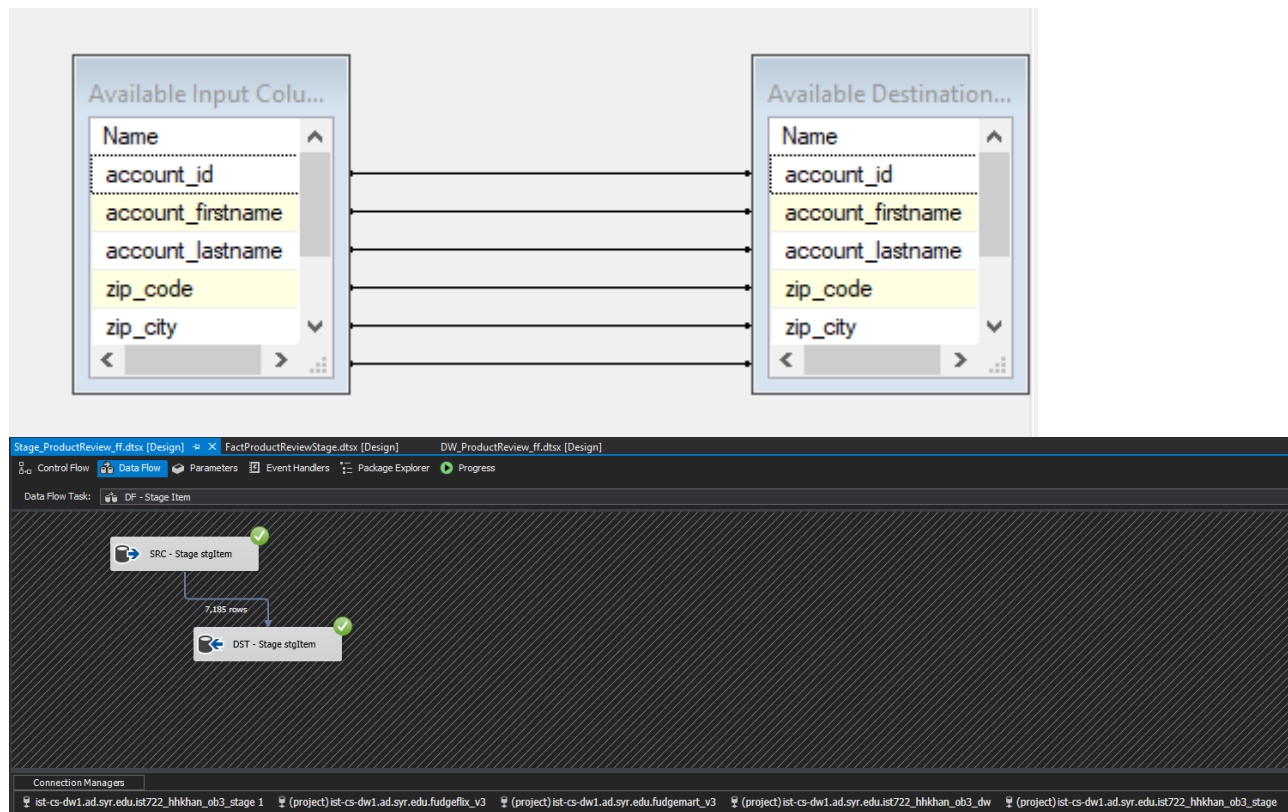
Stage_ProductReview_ff Control Flow: Similarly, we used the Execute SQL Task and two data flow controls to stage Product review from Fudgeflex where we have truncated Stage User and Stage Item in the Execute SQL Task control by using the following Truncation statement. The source to target mapping is also shown



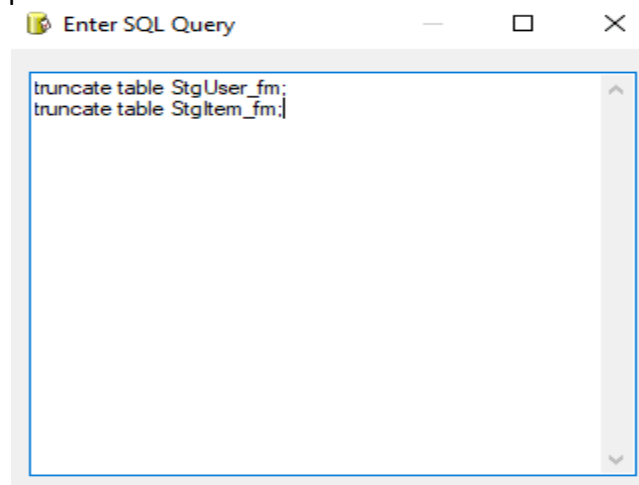
Stage_ProductReview_ff Data Flow: For Stage Item in the SRC we used the following SQL SELECT statement to select and join what we wanted from source (Fudgeflex)

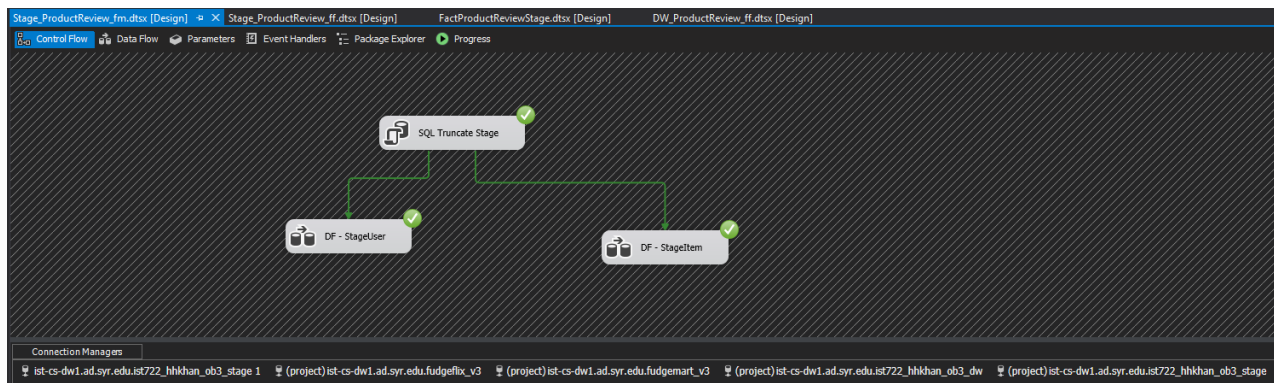


Then in the DST we are connected to our stag and specifically table stgUser_FF table to push the data. See the source to target data flow mapping

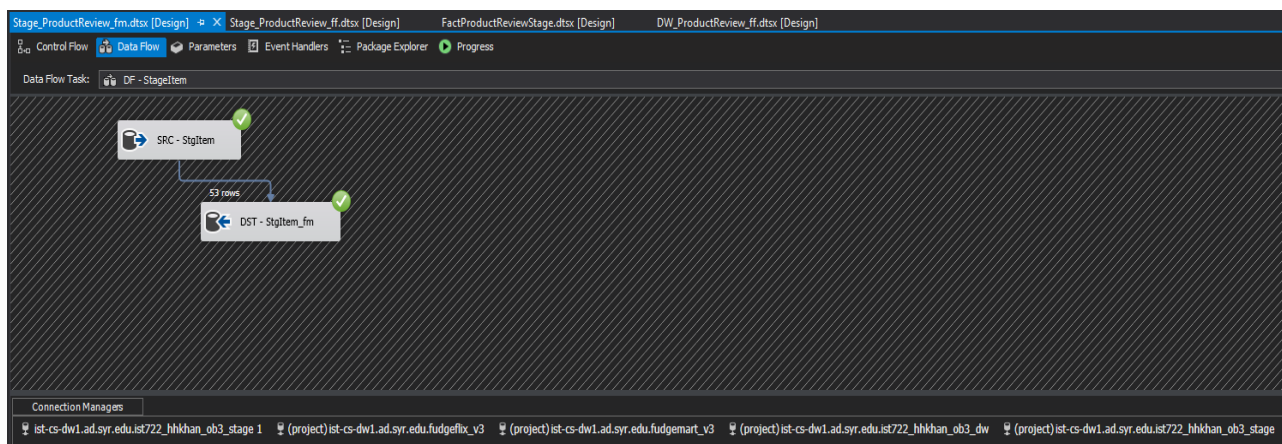
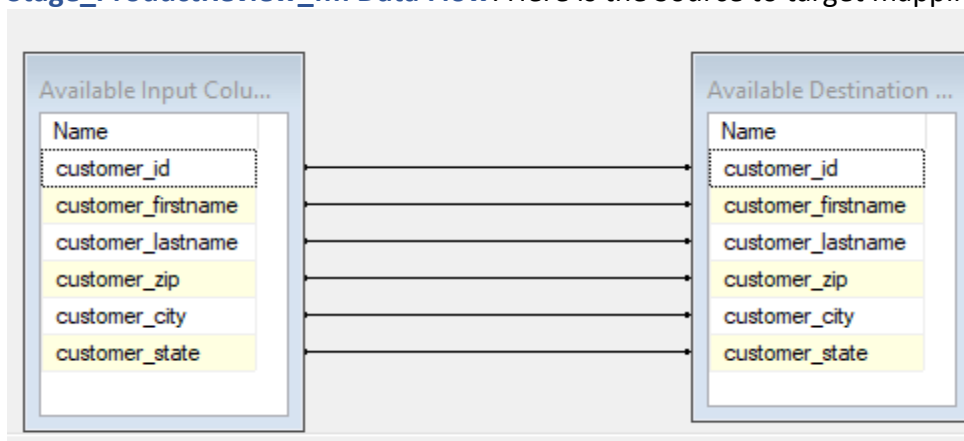


Stage_ProductReview_fm Control Flow: We used the same process as Stage_ProductReview_FF for Stage_ProductReview_FM where we used one Execute SQL Task and two data flow controls for Stage User and Stage Item and truncated the two tables by using the following truncation statement in the Execute SQL Task and here is our source to target screen capture for this process



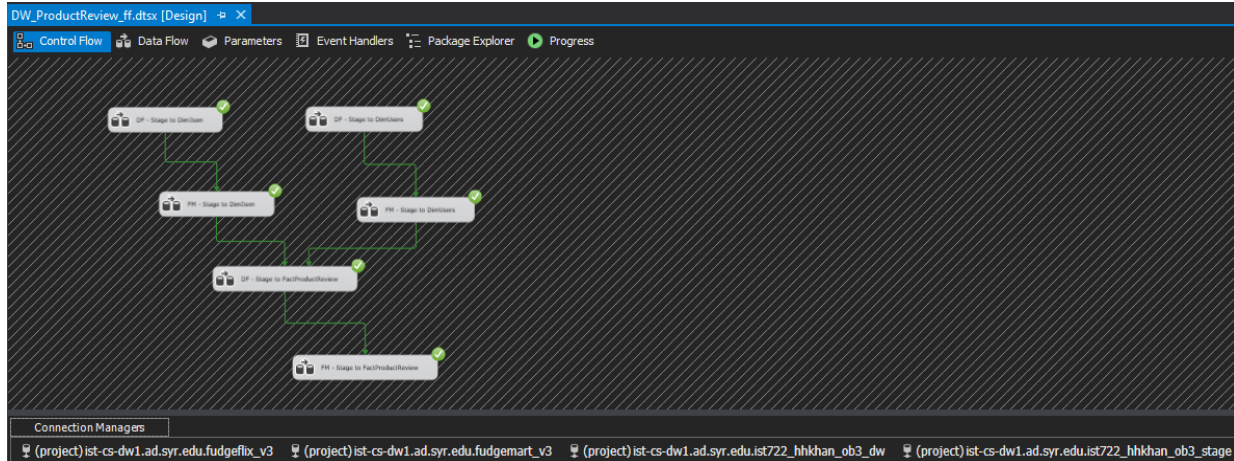


Stage_ProductReview_fm Data Flow: Here is the Source to target mapping in the data flow

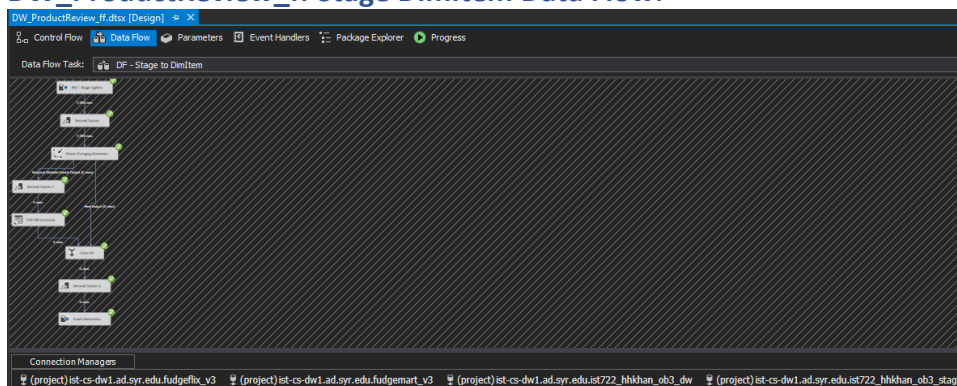


DW_ProductReview_FF Control Flow: This is the process that put everything together.

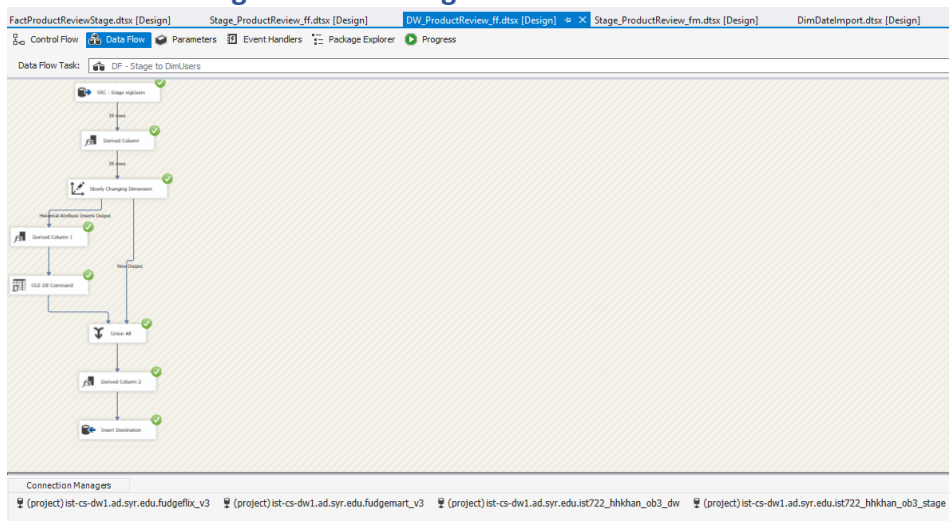
- Fudgeflex DimItem and DimUser were loaded to the data warehouse
- Fudgemart DimItem and DimUser were loaded to the data warehouse
- Fudgeflex Fact Table was loaded to the data warehouse
- Fudgemart Fact Table was loaded to the data warehouse



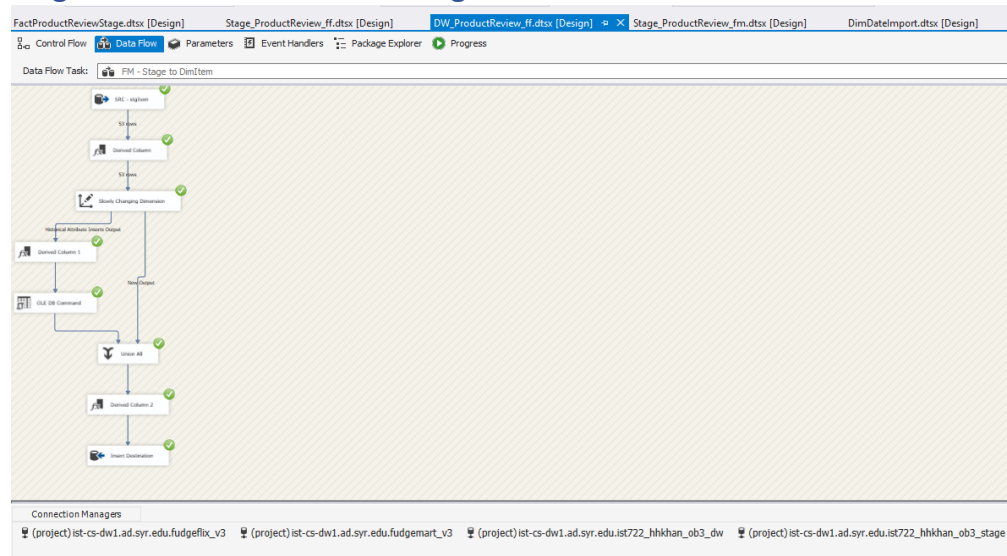
DW_ProductReview_ff Stage DimItem Data Flow:



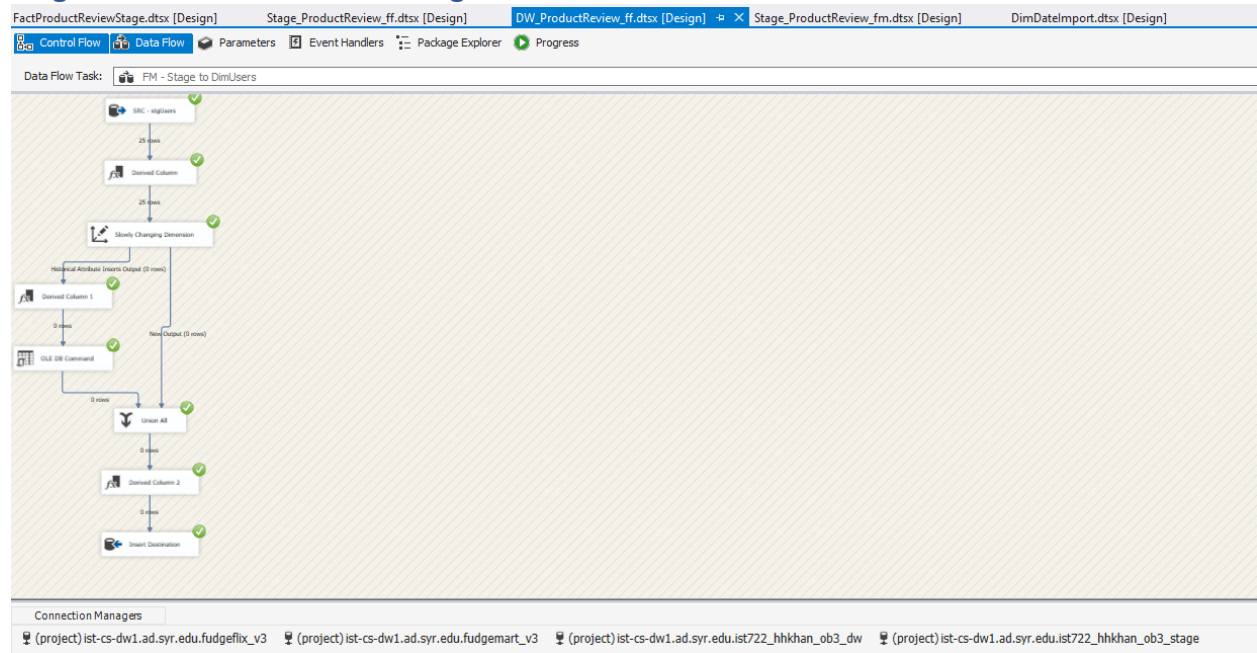
Data flow for Stage DimUser Fudgeflex



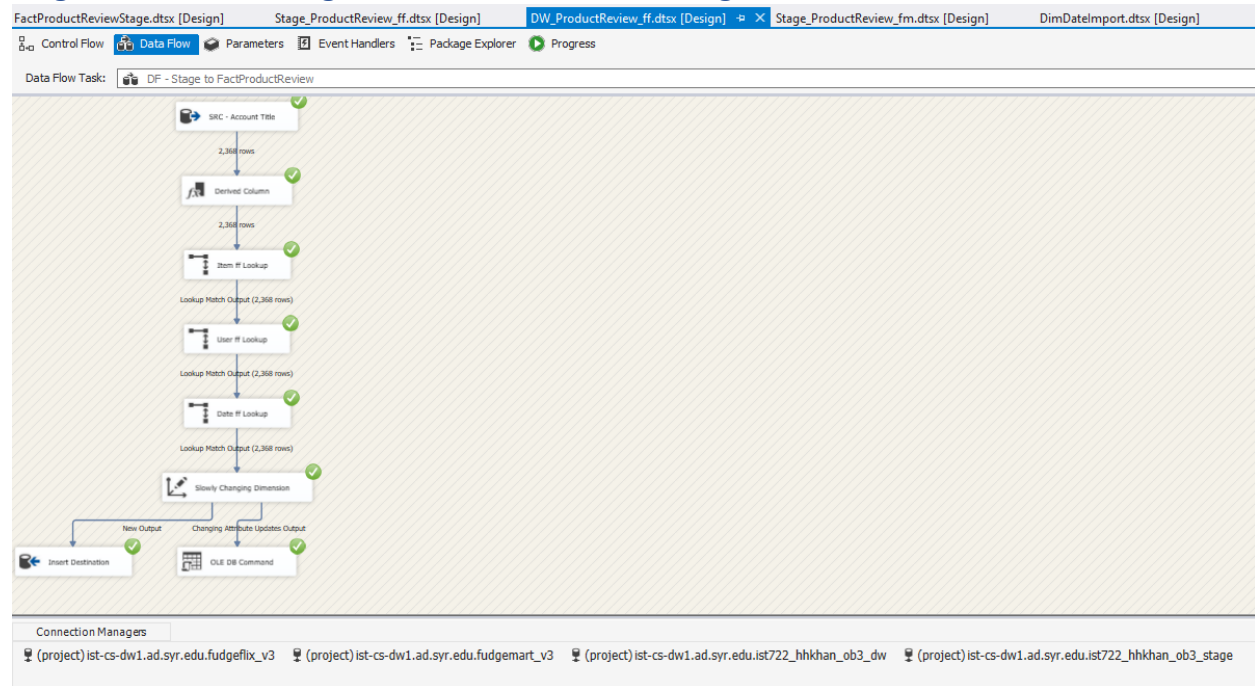
Stage DimItem Data Flow for Fudgeflex



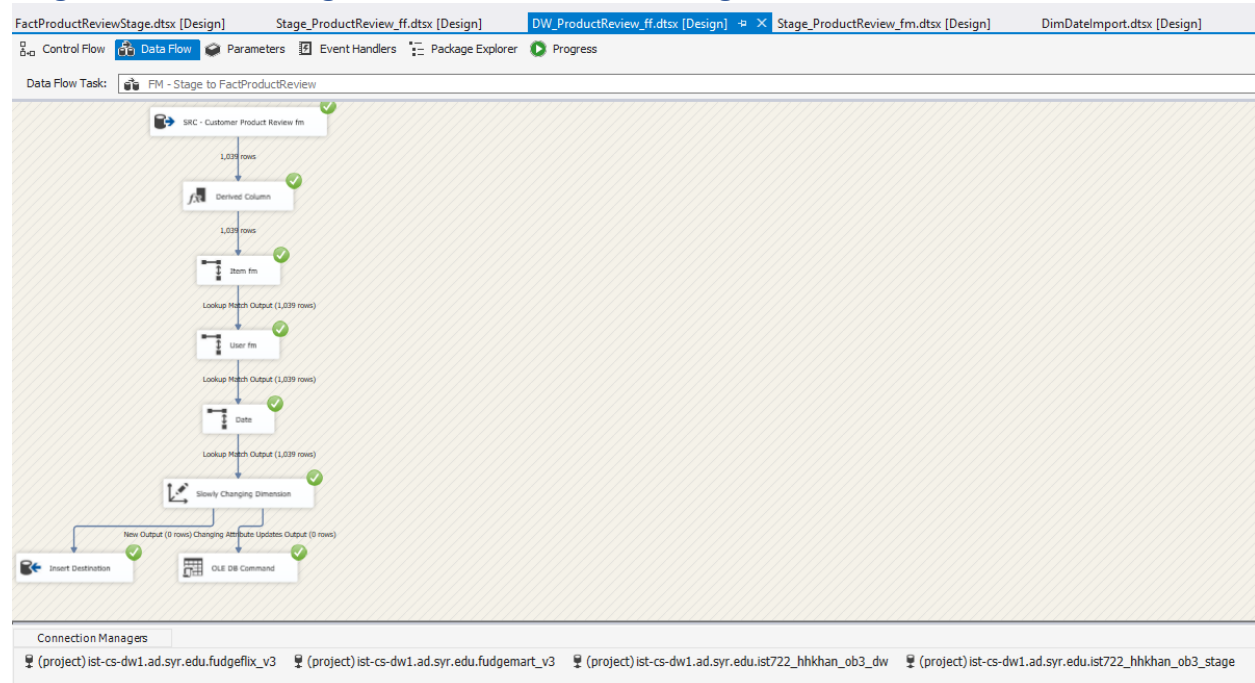
Stage DimUser Data Flow for Fudgemart



Stage Data Flow for Stage FactProductReview for Fudgeflex



Stage Data Flow for Stage FactProductReview for Fudgemart



Data Quality and Survivorship Rules

Identification of data types in both Fudgeflex and Fudgemart databases was key to ensuring the tables were properly staged. Some of the issues when loading the stage table data into the data warehouse included data type mismatches, and incorrect expression syntax.

Entering the correct columns and expressions alongside the SQL code in MS Server remedied data mismatches as it could be ensured expressions were built out correctly.

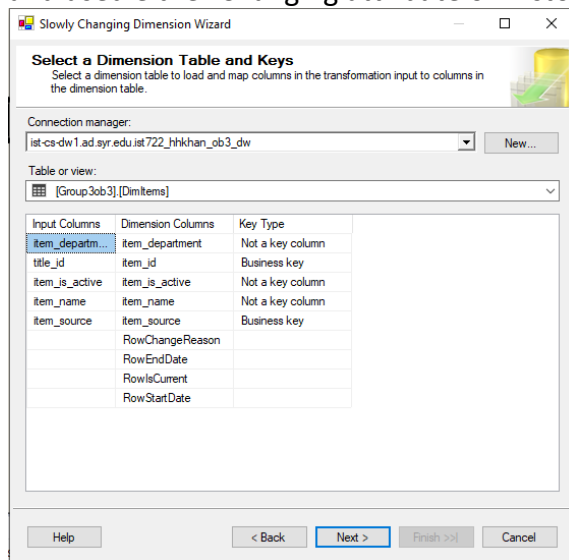
For ProductReview Data flows for both Fudgfex and Fudgemart what we most frequently used is

- Derived Column: to transform columns

Example:

| Derived Column Name | Derived Column | Expression | Data Type | Length | Precision | Scale | Code Page |
|---------------------|---------------------|---------------------------------------------------------|--------------------------|--------|-----------|-------|-----------|
| item_source | <add as new column> | (DT_WSTR,20)"fudgefix" | Unicode string [DT_WS... | 20 | | | |
| item_is_active | <add as new column> | (DT_BOOL)(title_bluray_available title_dvd_availa... | Boolean [DT_BOOL] | | | | |
| item_department | <add as new column> | (DT_WSTR,20)title_type | Unicode string [DT_WS... | 20 | | | |
| item_name | <add as new column> | (DT_WSTR,200)title_name | Unicode string [DT_WS... | 200 | | | |

- Slowly Changing Dimensions where we mapped the existing and transformed columns and used either Changing attribute or Historical attribute types. Example:



- Lookups where we looked up and mapped columns. Etc. Example:

