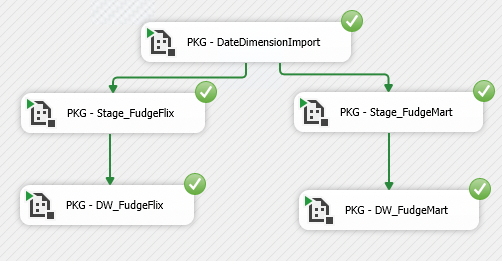
Group 5

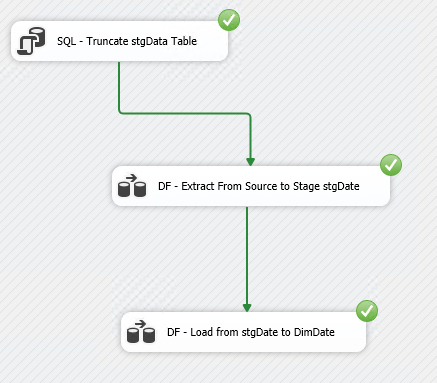
**Overall ETL Logic**

The ETL process is composed of 5 smaller packages, controlled by 1 master package. The role play date dimension is loaded first. Then, FudgeFlix and FudgeMart data is loaded in parallel. For both lines of business, the data is extracted from source to stage, then it is loaded from stage to the data warehouse in unified facts and dimensions.



**Date Dimension**

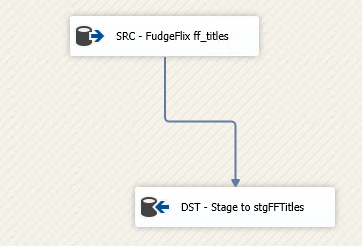
The date import model uses the same pattern that was used in class Assignment 6.

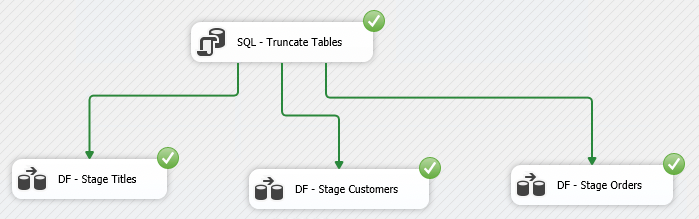


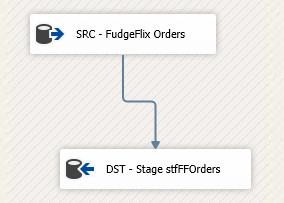
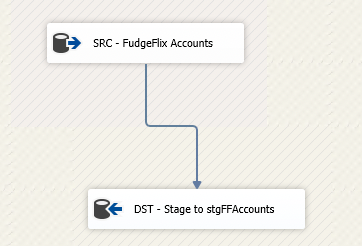
ImageImage

**Staging FudgeFlix**

Data is moved from source to staging tables without transformation. All the stage tables are truncated at the start of the script, then loaded in parallel.



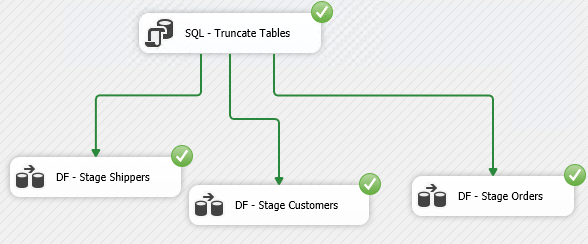


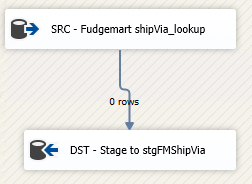
ImageImage

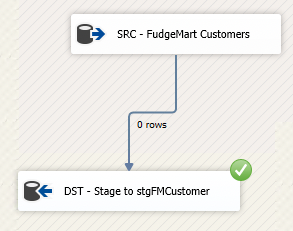
Image

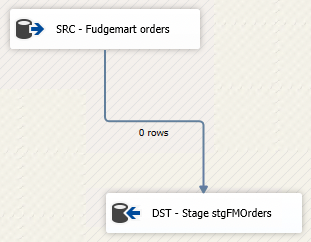
**Staging FudgeMart**

FudgeMart’s process is identical to FudgeFlix.

****

****

****

****

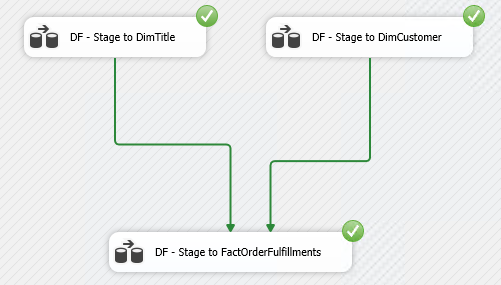
**Image**

**Image**

**Image**

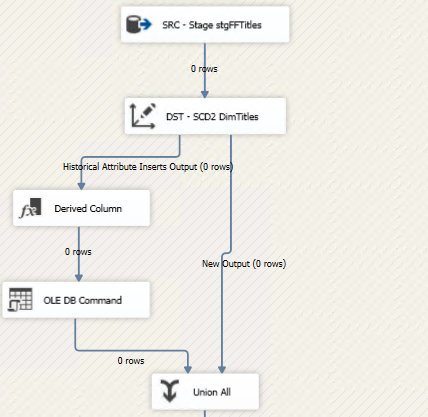
**Loading FudgeFlix**

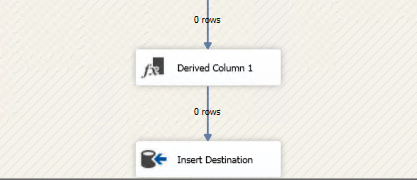
Data is transformed, as needed, and moved to the dimensions first, then the fact tables.



**Loading DimTitles from FudgeFlix**

Titles is loaded into an SCD Type 2 with no transformations necessary. Title ID is used as a business key. Titles are only sourced from FudgeFlix.

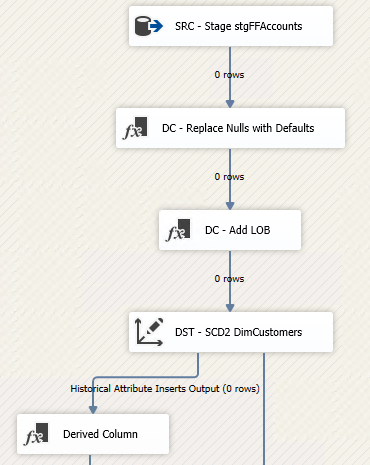


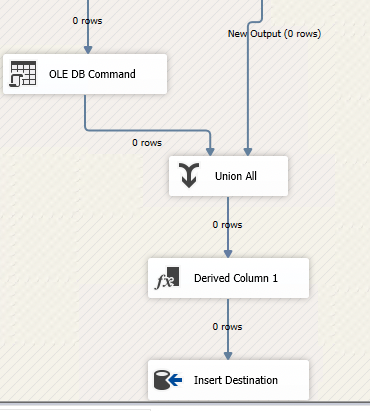


**Loading DimCustomer from FudgeFlix**

DimCustomers is a single dimension containing FudgeFlix and FudgeMart customers. FudgeFlix calls these accounts, but we have standardized and unified the naming convention to Customers. Blank Address, City, State or Zip codes are replaced with an ‘unknown value’ string. Phone and Fax are not applicable to FudgeFlix customers, and are all replaced with ‘unknown value’ strings. We added a column for line of business, to differentiate between customers of either system. We considered merging the two together. However, after analysis, data mismatches were present in several columns (street address, city, postal code, name). We could not guarantee that an email address used in FudgeFlix would map to the exact same customer in FudgeMart, as email addresses could be shared between different individuals. Furthermore, the source systems have no mechanism to update themselves should the data change in the other. We recommend implementing an MDM strategy to merge the names together. When that happens, the LOB column can either be deprecated or used to track which accounts were merged together during the transition.

Customers are loaded via a SCD Type 2. The FudgeFlix account\_id combined with the LOB is used as a business key. This ensure that if FudgeMart has the same account Ids, they will be treated as separate entities.

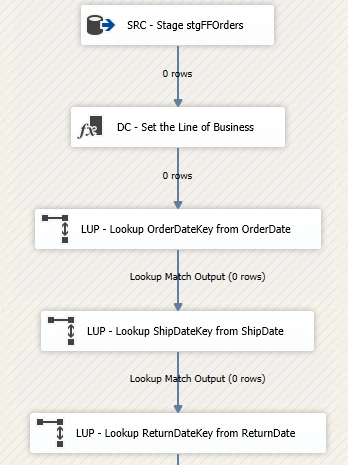


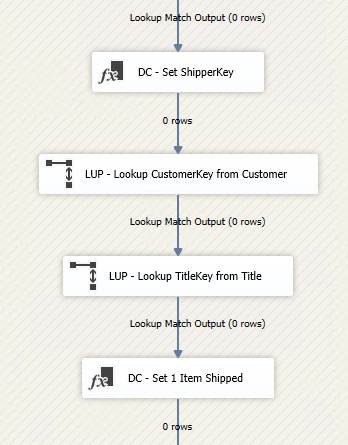
Image

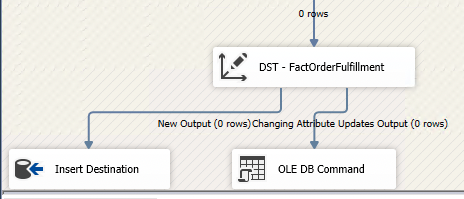
**Loading FactOrderFulfillments from FudgeFlix**

Like the customers dimension, FactOrderFulfillments are combined between the two businesses. A column has been created to track which line of business the order has come from. The order date, ship date and return date are all transformed to the DimDate key. FudgeFlix has no shipper option. We set the ShipperKey to -2 for all entries. The -2 entry in DimShipper is keyed expressly to indicate FudgeFlix as the shipper. The customer id is transformed into the DimCustomer key, and the Title is transformed into the DimTitle key. Quantity is more of a FudgeMart trait, as every order in FudgeFlix is shipped separately. As such, we set the Quantity to 1 for every order. Finally, we derive the shipping lag as the difference in days between queue date and ship date, while return lag is the difference between shipped date and returned date.

OrderFulfillments are loaded via a SCD Type 2. The FudgeFlix at\_id combined with the LOB is used as a business key. This ensure that if FudgeMart has the same order id, they will be treated as separate entities.

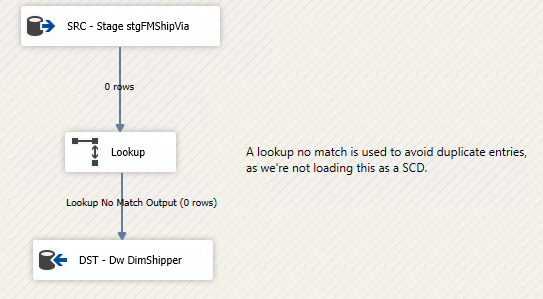






Image

**Loading DimShipper from FudgeMart**

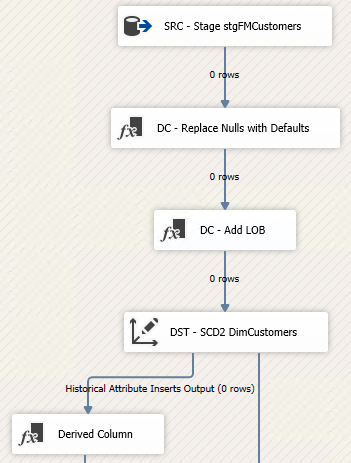
Shippers is a simple lookup field in the FudgeMart source system. As we have no business keys aside from name, we can do a simple load. Initially, we found this resulted in duplicate shipper names when the ETL script was re-executed. A lookup element was added. If the shipper is already in the data warehouse, we skip it, and only load non-matching values.

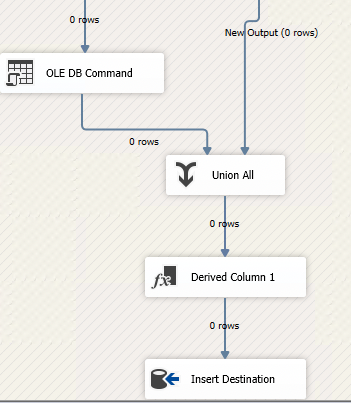
Image

**Loading DimCustomer from FudgeMart**

The loading of FudgeMart customers is similiar to FudgeFlix. As email and fax are not required fields in FudgeMart, any blank values are replaced with ‘unknown value’ text. A line of business column is added as well.

Customers are loaded via a SCD Type 2. The FudgeMart customer\_id combined with the LOB is used as a business key. This ensure that if FudgeFlix has the same account Ids, they will be treated as separate entities.



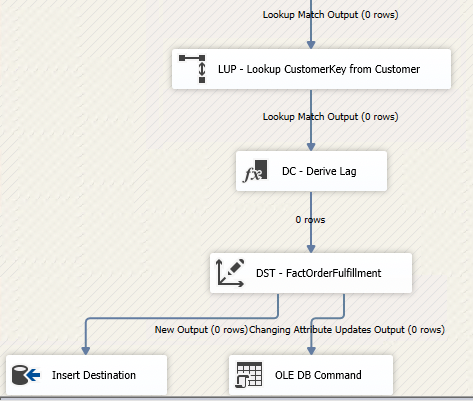
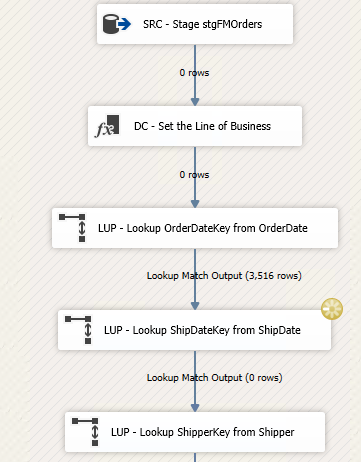


Image

**Loading FactOrderFulfillment from FudgeMart**

The loading of FudgeMart orders is similar to FudgeFlix. A column has been created to track which line of business the order has come from. The order date and ship date are transformed to the DimDate key. The shipper is transformed into the DimShipper key, and the customer is transformed into the DimCustomer key. Quantity was derived from the source system by doing a join with summation on the order details table. Finally, we derive the shipping lag as the difference in days between order date and ship date.

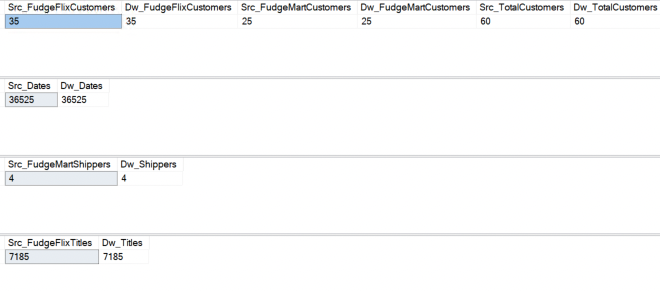
OrderFulfillments are loaded via a SCD Type 2. The FudgeMart order\_id combined with the LOB is used as a business key. This ensure that if FudgeFlix has the same order id, they will be treated as separate entities.



Image

**Initial Data Load Verification**

In testing the initial data load, we counted up the number of records in each source system and compared it to the values in the data warehouse. We looked at the numbers for each line of business as well as the total values. We factored in that the Dimension tables have placeholder ‘unknown’ values loaded as part of the Data Warehouse creation scripts. The SQL code for this verification has been included in the deliverables.



Image

Image