Group 1 ETL Documentation

**Date Dimension Import:**

The first step in the ETL Process after generating the script was to add the Date Dimension. For this, the same process as the homework was followed to import the date from the External Source.

**Control Flow:**

Diagram

Description automatically generated

**Data Flows:**

Source to Stage

**Text, application, chat or text message

Description automatically generated**

Stage to Dim Table

**Diagram

Description automatically generated**

Next, we staged the Customer, Product, and Sales dimensions from the external sources (FudgeFlix and FudgeMart) to the stage tables.

**Control Flows (Product, Customer, Sales)**

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated

**Data Flows:**

Each company’s dimensions were merged when going from the source to the stage table. That is why The Execute SQL Task connects to both the mart and flix named components. For example, the fudgeMart customers and fudgeFlix accounts were merged to one source table named “stgFudgeCustomer.”

The data flows for each of these 6 inputs followed the same structure. The source was from the table within either fudge flix or fudge mart. Next, a derived column was added with the column “Source” added. For all flows coming from fudgemart, a “FM” was assigned. For fudgeflix, a “FF” was assigned. Additionally, assumptions were added in the derived columns. For fudgeflix on the products side, product\_wholesale\_price, which doesn’t exist naturally, was assigned a value of $2. When staging from the sources to the staged Sales table, a derived column “Quantity” where quantity = 1, was assigned in the fudgeflix flow where fudgemart had the variable order\_qty.

To avoid redundancy, only the data flow for moving fudgeflix product source data to the merged stgFudgeProduct table is shown. The other 5 data flows follow the same mechanism.

Diagram

Description automatically generated

Shown below is an example of the derived columns added within the data flow. This example represents the derived columns for the fudgeflix sales data as it gets merged to the stgFudgeSales table with the sales data from fudgemart.

Graphical user interface, application

Description automatically generated

**Stage Tables to Data Warehouse**

Because we have already merged the business process in the stage tables, we only need one fact table for our Data Warehouese. This is the FactSale table. The measures are Revenue (retail\_price/plan\_price \* quantity) and Costs (wholesale\_price \* quantity).

**Control Flow:**

Diagram

Description automatically generated

First, we move data from the stgFudgeProduct table to the DimProduct table and do the same with the stgFudgeCustomer table to the DimCustomer table. Once that data is loaded in, we stage to the FactSales table.

**Data Flows:**

**DimProduct –**

**Diagram

Description automatically generated**

**DimProduct Derived Columns: Renaming columns to match destination**

**Graphical user interface

Description automatically generated with medium confidence**

**DimCustomer –**

**Diagram

Description automatically generated**

**Derived Columns: Match naming conventions and create CustomerName as composite of firstname and lastname.**

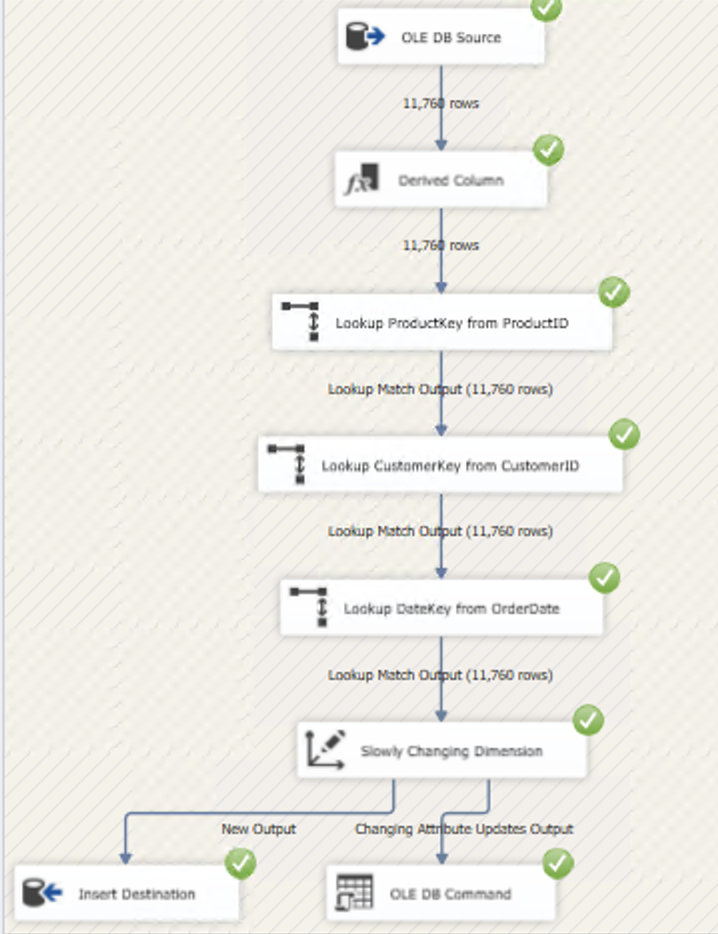
**Graphical user interface

Description automatically generated**

**FactSale – ProductKey looked up from ProductID and Source**

**CustomerKey from CustomerID (ID for each company are unique, no source needed)**

**DateKey from OrderDate**

****

**Derived Columns- Add Revenue, Costs, and Quantity.**

**Graphical user interface, table

Description automatically generated with medium confidence**