ETL Documentation

This ETL documentation is meant to clarify the ETL process from the FudgeMart and FudgeFlix OLTP databases into our data warehouse. The facts that we wanted to track were the total sales for the company and the reviews customers have provided for FudgeMart products. There were six different packages created for staging the data and loading the data into the data warehouse, as well as one package that combined all packages to streamline the ETL process. The ETL packages that were created included:

* **DateDimensionImport** - Extracted date data from ExternalSources2, staged the date data in the stage database, and loaded the data into the data warehouse
* **Stg\_FactTotalSales** - Extracted the products, customers, and order details from the FudgeMart database and extracted the account, plan, and account billings from the FudgeFlix database.
* **DW\_FudgeMart\_TotalSales** - loaded the FudgeMart products from stage into the DimProducts and loaded the FudgeMart Customers from stage to DimCustomer. Also derived the total sales from the order detail by multiplying the quantity sold by the product’s retail price and inserted the sales record into the FactTotalSales table by looking up the CustomerKey and the ProductKey.
* **DW\_FudgeFlix\_TotalSales** - loaded the FudgeFlix plans from stage into the DimProducts and loaded the FudgeFlix Customers from stage to DimCustomer. Also inserted the billing record into the FactTotalSales table by looking up the CustomerKey and the ProductKey.
* **Stg\_FactFudgemartReviews** - Extracted the customer review from the FudgeFlix database and staged the data.
* **DW\_FactFudgemartReviews** - Loaded the FudgeMart reviews from the staging environment into the data warehouse.
* **Package** - Combined all of the packages in order to streamline the ETL process in one step.

## Source-To-Target Map

**DimProduct**

|  |  |  |
| --- | --- | --- |
| **FudgeMart** | **FudgeFlix** | **Data Warehouse** |
| Derived Identity | Derived Identity | ProductKey |
| fm\_products.product\_id | ff\_plan.plan\_id | ProductID |
| Default “FudgeMart” | Default “FudgeMart” | Subsidiary |
| fm\_products.product\_name | ff\_plan.plan\_name | ProductName |
| fm\_vendors.vendor\_name | Default “FudgeFlix Plan” | SupplierName |
| fm\_products.product\_department | ff\_plan.plan\_name | ProductCategory |
| fm\_products.product\_is\_active | ff\_plan.plan\_current | Discontinued |
| SCD Type 2 Active Row | SCD Type 2 Active Row | RowIsCurrent |

**DimCustomer**

|  |  |  |
| --- | --- | --- |
| **FudgeMart** | **FudgeFlix** | **Data Warehouse** |
| Derived Identity | Derived Identity | CustomerKey |
| fm\_customer.customer\_id | ff\_accounts.account\_id | CustomerID |
| Default “FudgeMart” | Default “FudgeFlix” | Subsidiary |
| fm\_customer.customer\_email | ff\_accounts.account\_email | CustomerEmail |
| fm\_customer.customer\_firstname | ff\_accounts.account\_firstname | CustomerLastName |
| fm\_customer.customer\_lastname | ff\_accounts.account\_lastname | CustomerFirstName |
| Derived | Derived | CustomerNameFirstLast |
| fm\_customer.customer\_city | ff\_zipcodes.zip\_city | CustomerCity |
| fm\_customer.customer\_state | ff\_zipcodes.zip\_state | CustomerState |
| fm\_customer.customer\_zip | ff\_accounts.account\_zipcode | CustomerZipCode |
| fm\_customer.customer\_phone | “N/A” | CustomerPhoneNumber |
| SCD Type 2 Active Row | SCD Type 2 Active Row | RowIsCurrent |

**FactTotalSales**

|  |  |  |
| --- | --- | --- |
| **FudgeMart** | **FudgeFlix** | **Data Warehouse** |
| Lookup product\_id | Lookup plan\_id | ProductKey |
| Lookup customer\_id | Lookup account\_id | CustomerKey |
| Lookup order\_date | Lookup ab\_date | TransactionDateKey |
| fm\_order\_details.order\_id | ff\_account\_billing.ab\_id | TransactionID |
| Default “FudgeMart” | Default “FudgeFlix” | Subsidiary |
| Derived fm\_order\_details.order\_qty \* fm\_products.product\_retail\_price | ff\_account\_billing.ab\_billed\_amount | SoldAmount |
| fm\_order\_details.order\_qty | Default 1 | SoldQuantity |
| fm\_products.product\_retail\_price | ff\_plan.plan\_price | UnitPrice |

**FactFudgeMartReviews**

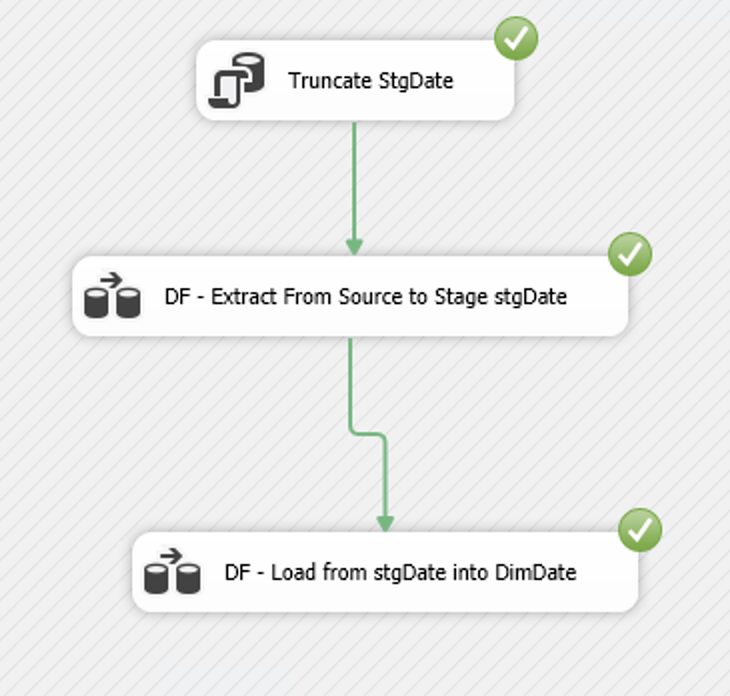
|  |  |
| --- | --- |
| **FudgeMart** | **Data Warehouse** |
| Lookup product\_id | ProductKey |
| Lookup customer\_id | CustomerKey |
| Lookup fm\_customer\_product\_reviews.review\_date | ReviewDateKey |
| fm\_customer\_product\_reviews.review\_stars | ReviewStars |

## Date Dimension Import

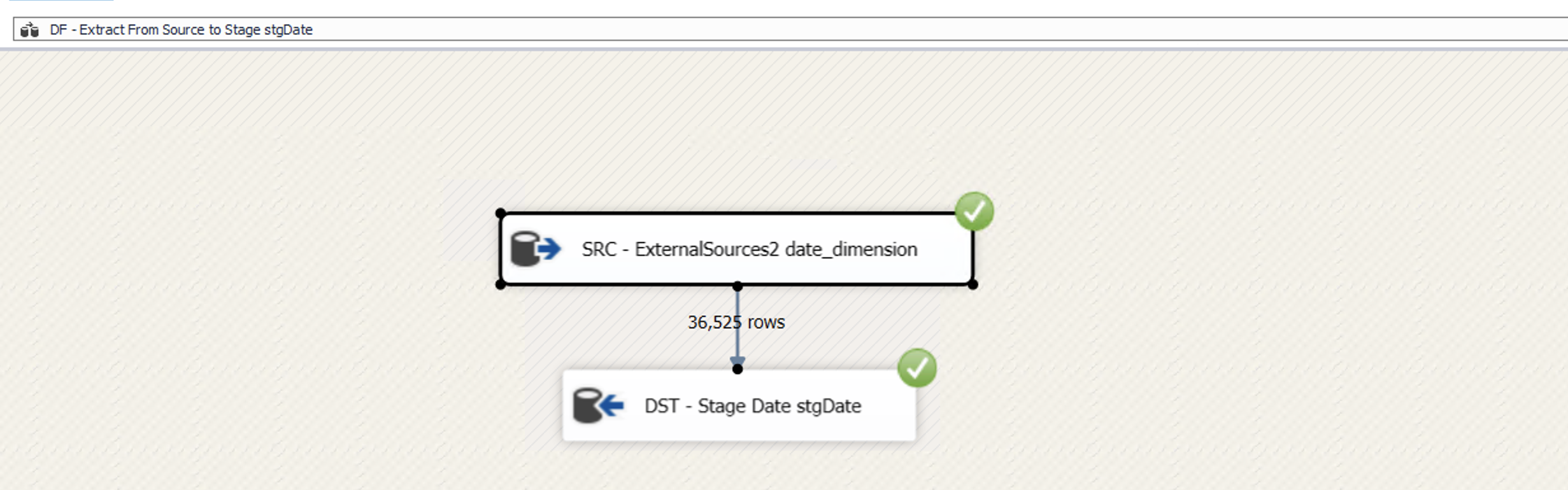
The date dimension import included a data flow task that truncated the stgDate table, and two data flow tasks: extracting the data from the source and loading the data from the stage to the data warehouse. The data was extracted “as-is” into the staging environment from the ExternalSources2 database. The data was then transferred from the stage environment into the data warehouse as a Type 1 slowly changing dimension, with the DateKey as the business key. Two columns were derived to change the data types:

* Year was cast from type int to type smallint
* DayOfYearUSA was cast from type int to type smallint

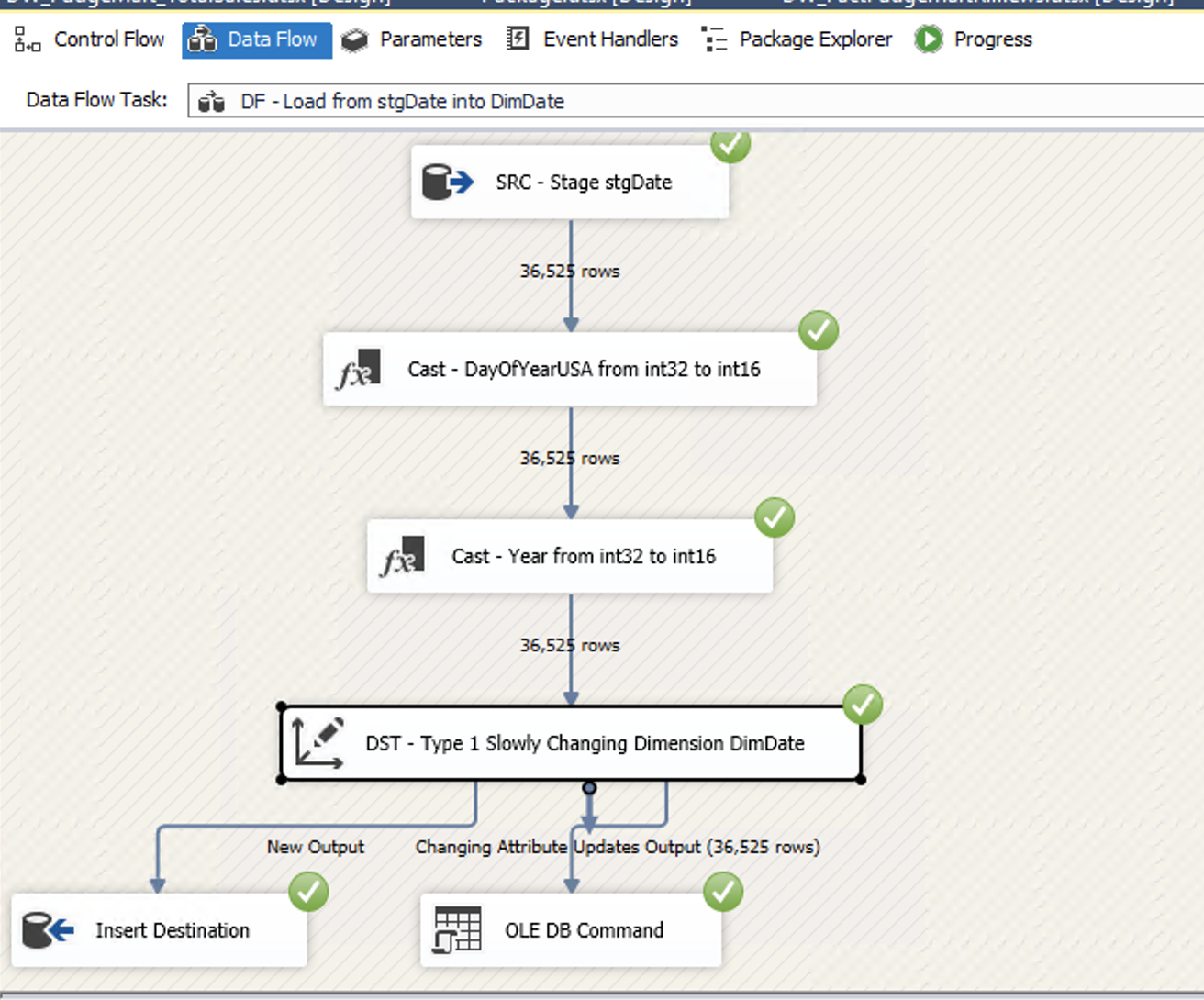
**Date Dimension Control Flow**



**Source to Stage Data Flow**



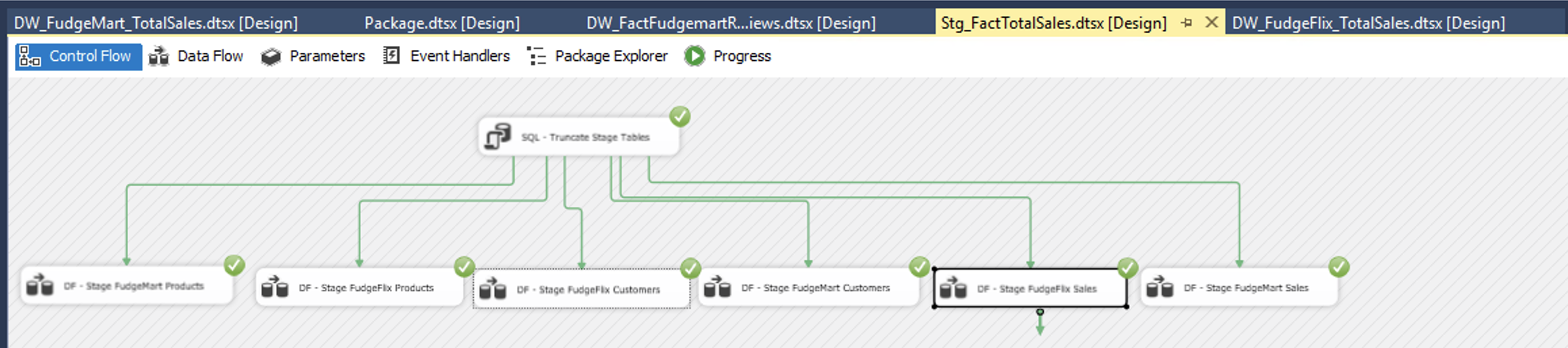
**Stage to Data Warehouse Data Flow**



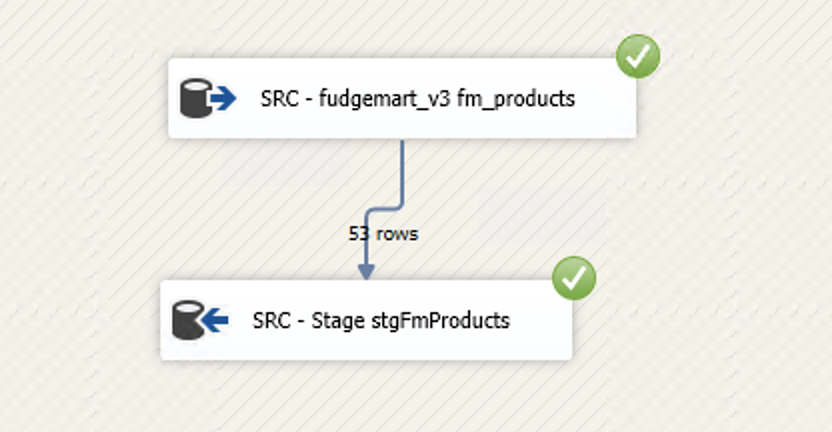
## Stage Total Sales

The stage total sales package included a data flow task that truncated each of the stgFmProducts, stgFmCustomers, stgFmSales, stgFfProducts, stgFfCustomers, and stgFfSales tables. There were also six data flow tasks which extracted the Fudgemart Customer, Products, and Order Details, and the FudgeFlix plans, accounts, and account billings. Additionally, an inner join was conducted on the FudgeMart order details and the FudgeMart orders to determine the order\_date. The product price was joined in the initial extract with the order details so the sold amount in the fact table could be later calculated. The same occurred with the account billing details and the plan price. The data for each of the dimensions and facts were stored directly in the stage database, with no additional processing or derived columns.

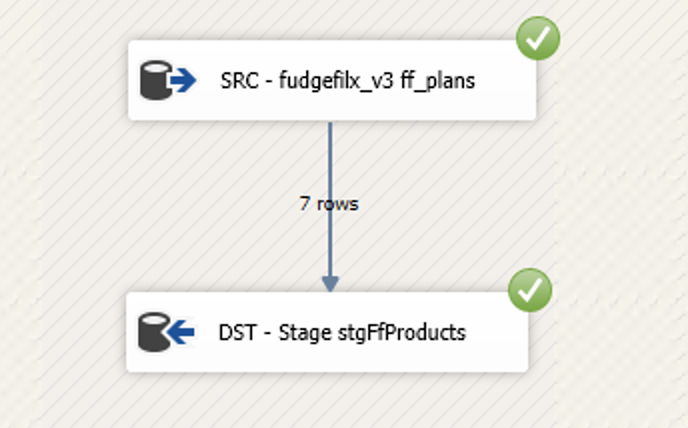
**Stage Total Sales Control Flow**

****

**Stage FudgeMart Products**



**Stage FudgeFlix Products**

****

## Data Warehouse Fact Total Sales (FudgeMart)

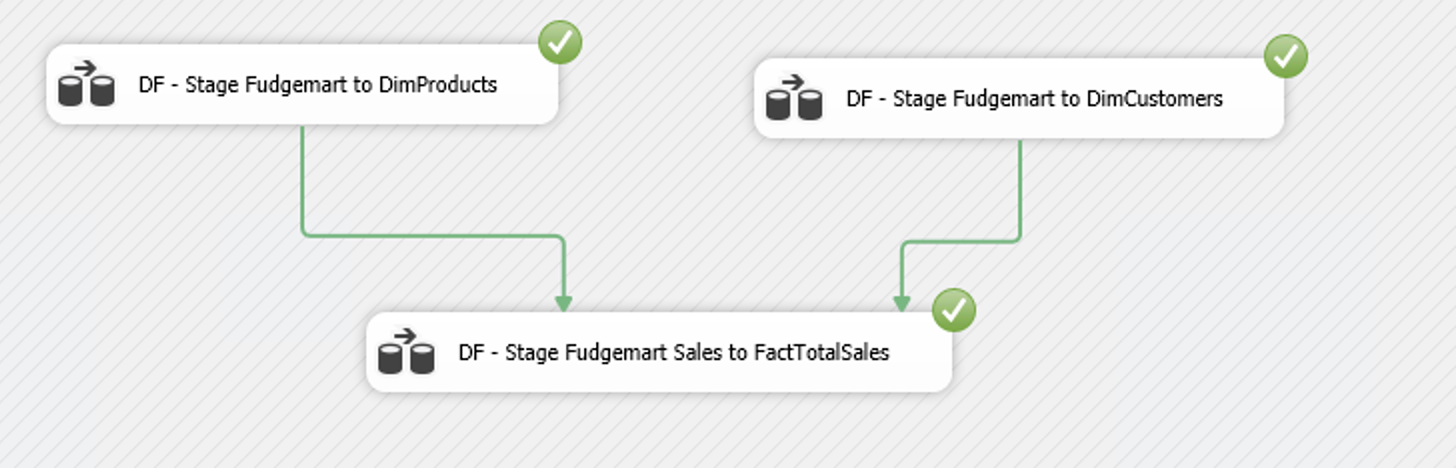
The packages to load all the sales from FudgeMart and FudgeFlix into the data warehouse were created separately to deal with any issues that arose one at a time. First, the FudgeMart Products and theFudgeMart Customers were loaded into the Customer and Product dimensions. In each of the data flows, a new column, Subsidiary, was derived and was given the default value of “FudgeMart”, indicating that within the dimensions, this product or customer was from the FudgeMart database. This allowed us to distinguish between potential overlapping business keys in either the product or customer dimensions when combined with the FudgeFlix customers and products.

When inserting into the product dimension, the Product Name and Product Category were cast to be of data type NVARCHAR(50), the Supplier Name was cast to data type NVARCHAR(20), and the Discontinued column changed the data type from type bit to either “Y” or “N” depending on whether the product had been discontinued or not. The product was then inserted into DimProduct as a Type 2 slowly changing dimension with the product id as the business key.

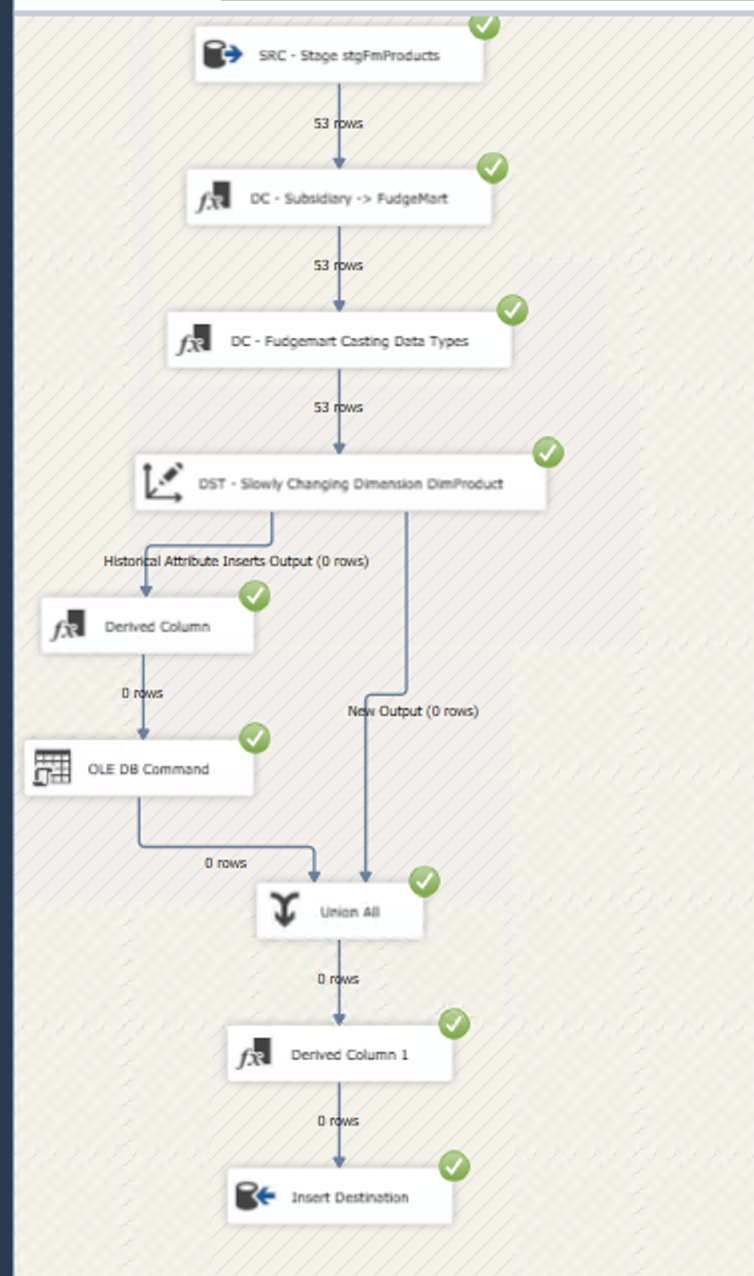
When inserting the the customer dimension, the customer’s first name and the customer’s last name were cast into data type NVARCAHR(50), the customer’s city was converted to all uppercase for consistency and then cast into data type NVARCHAR(50), the customer’s email was cast to data type NVARCHAR(200), the customer’s state was cast into NVARCHAR(12), the customer’s zip code was converted to NVARCHAR(20), the customer’s phone was converted to NVARCHAR(12). Additionally, the customer’s first and last names were combined and separated by a space to be in the format “First Last”. The customer was then inserted into the DimCustomer as a Type 2 slowly changing dimension with the customer id as the business key.

After both the Product and Customer dimensions had been loaded into the data warehouse, the total sales were inserted into the data warehouse. The total sales for FudgeMart calculated the sold amount by multiplying the product’s retail price by the ordered quantity, and cast the result as the money data type. The date key was looked up using the order date, the product key was looked up by finding the Product ID that was also from the same subsidiary, and Customer key was looked up by the Customer ID and the subsidiary. Finally, the sale was inserted into the FactTotalSales table as a Type I Slowly Changing Dimension, using the order\_id and the product key as the business keys.

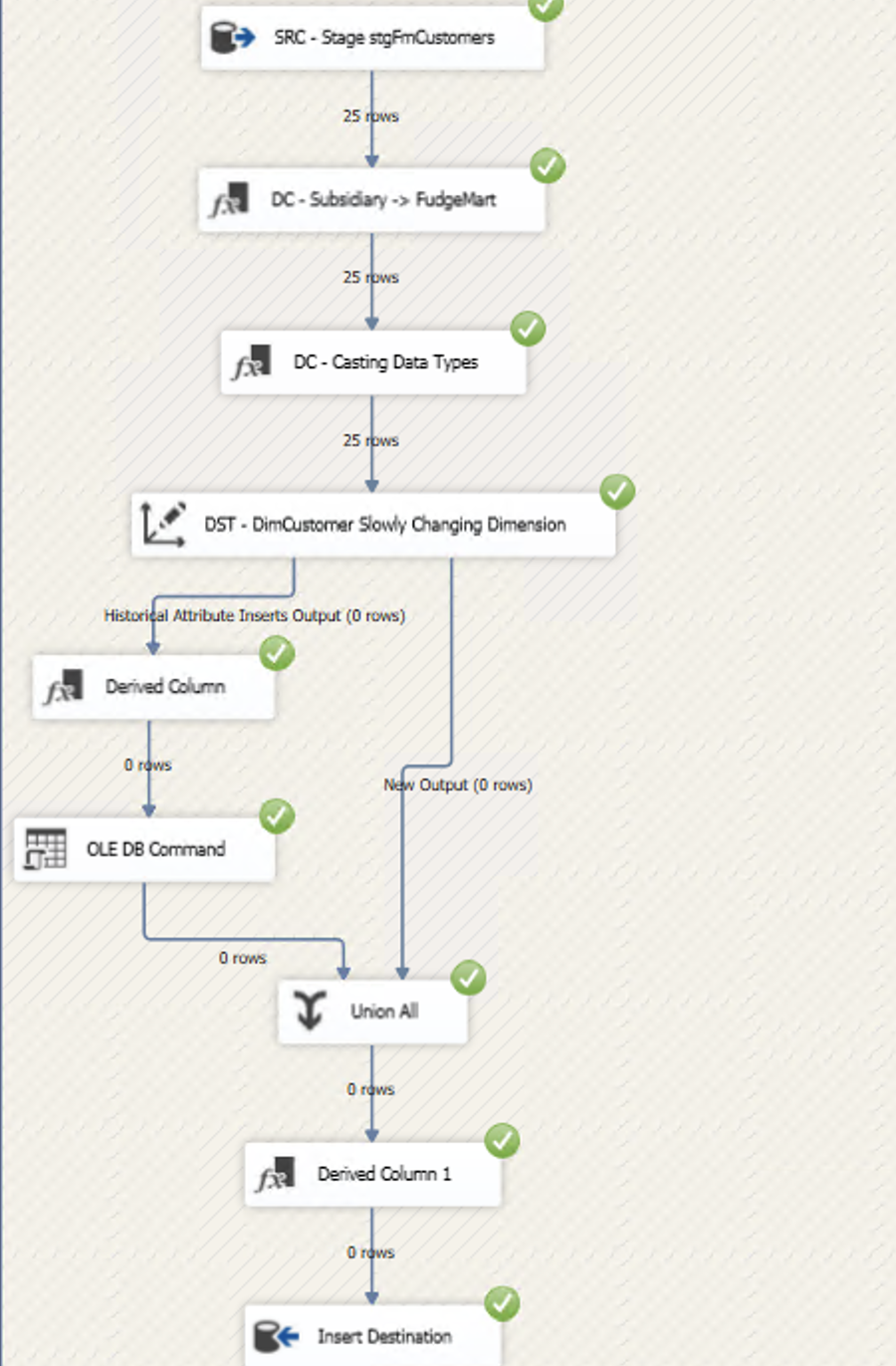
**Control Flow FudgeMart Total Sales to Fact Sales**



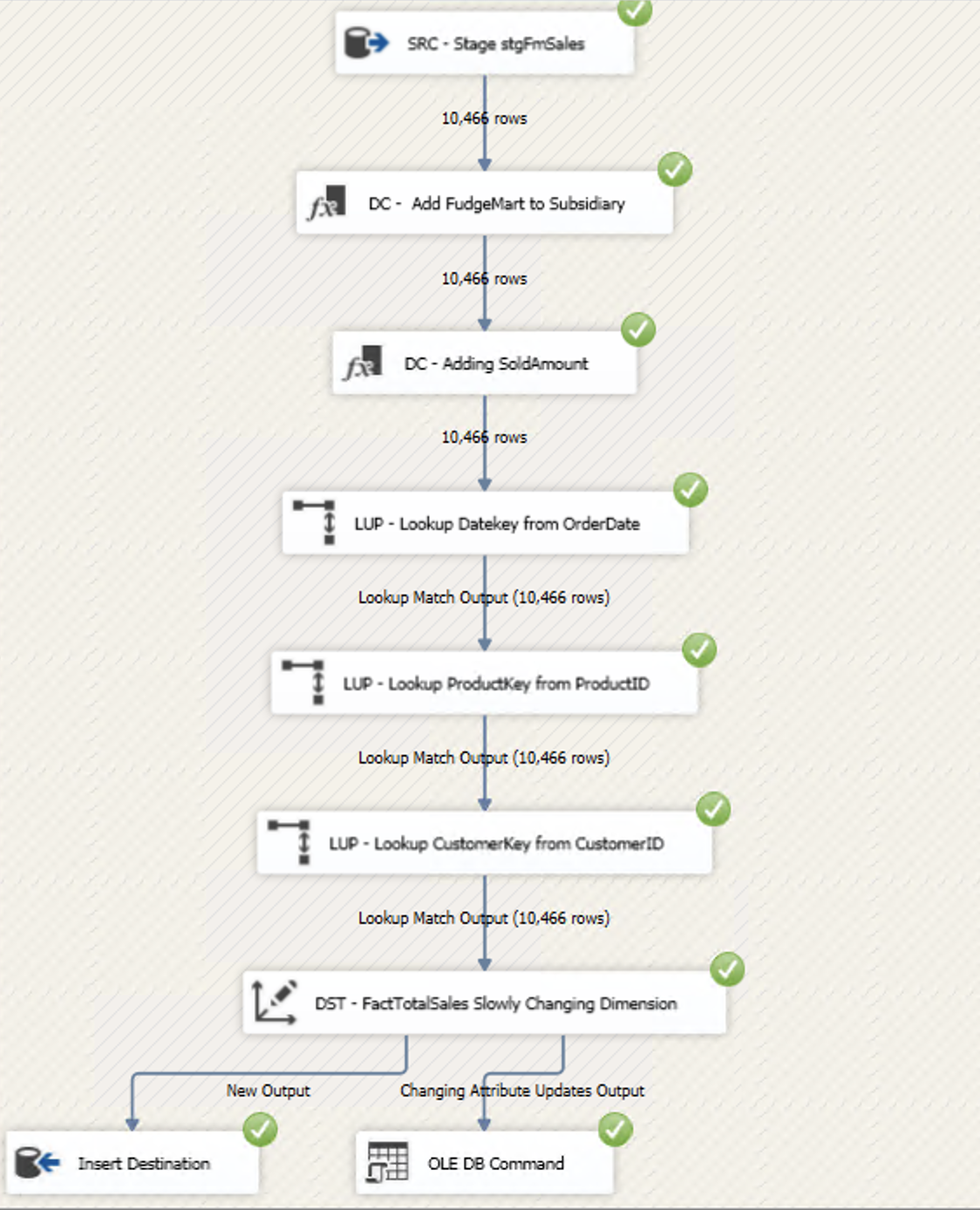
**Load Products Dimension From Staged FudgeMart Products**

****

**Load Customers Dimension From Staged FudgeMart Customers**

****

**Load Total Sales Fact From Staged Fudgemart Sales**



## Data Warehouse Fact Total Sales (FudgeFlix)

The package that loaded the FudgeFlix Sales into the Total Sales Fact table included three data flows that loaded the FudgeFlix Customers, loaded the FudgeFlix Products, and then loaded the FudgeFlix sales into the data warehouse. In each of the data flows, a column, Subsidiary, was created and given the default value of “FudgeFlix” indicating all of the data was arriving from the FudgeFlix subsidiary. Additionally, the CustomerPhone column was given the default value of N/A because there the phone number was not recorded for any FudgeFlix customers.

The data flow that added the FudgeFlix plans into the DimProduct also involved some data type cast conversion for the Plan name into NVARCHAR(50), and it would be the de facto “Product Name”, the “plan is current” column was converted from the bit data format into a “Y” or “N” depending on if the plan had been discontinued or not, and the Supplier Name columns was given the default value of “FudgeFlix Plan” because the sales were not directly from a supplier. Then the FudgeFlix plans were inserted into the DimProducts table as a Type 2 Slowly Changing Dimension.

The data flow that added the FudgeFlix Customers to the data warehouse cast the account firstname to NVARCHAR(50), account lastname to NVARCHAR(50), account state to NVARCHAR(12), account email to NVARCHAR(200), and account zip code to NVARCHAR(20). The zip\_city was converted to be all uppercase, and cast to NVARCHAR(50). Additionally, the account’’s first name and the account’s last name were combined and separated by a space to create a “First Last” name pattern for the account. After the columns were correctly cast and the

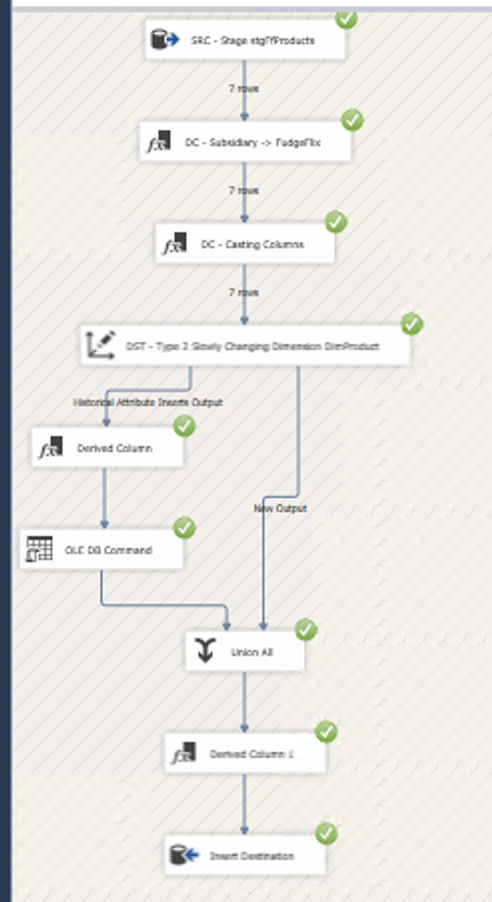
Derived columns were properly formatted, the FudgeFlix accounts were inserted into the DimCustomer table as a Type 2 Slowly changing dimension with the account\_id and the subsidiary acting as the business keys.

After both the FudgeFlix customers and products had been inserted into the data warehouse, the FudgeFlix sales were loaded into the data warehouse. The derived columns included Sold Amount, Quantity Sold, and Unit Price. The amount sold was derived as the account billed amount cast as the money data type, the quantity sold was given the default value of 1, and the Unit price was the plan price cast as the money data type. After the columns had been cast to the proper data type, the Date key was looked up using the ab\_date, the customer key was looked up using the account\_id where the subsidiaries matched, and the product key was looked up using the plan\_id where the subsidiaries matched. Finally, the FudgeFlix sales were inserted into the FactTotalSales Table as a Type 1 Slowly Changing Dimension, using the ProductKey and the ab\_id as the business keys.

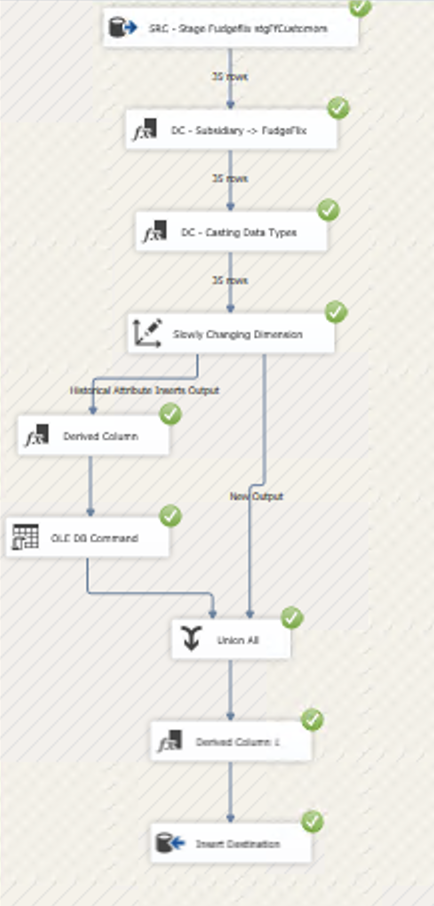
**Control Flow FudgeFlix Total Sale**

****

**Data Flow for FudgeFlix Plans to DimProducts**

****

**Data Flow FudgeFlix Accounts to DimCustomers**

****

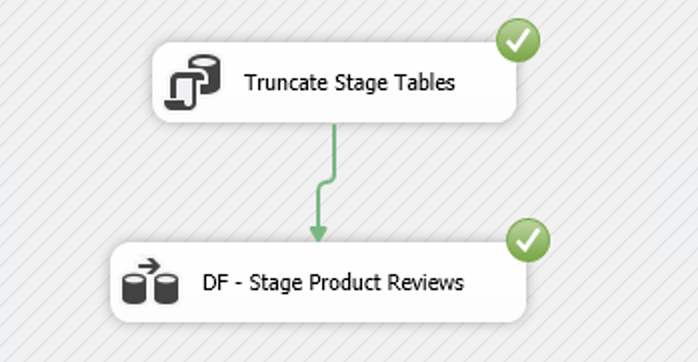
**Data Flow FudgeFlix Account Billing to FactTotalSales**

****

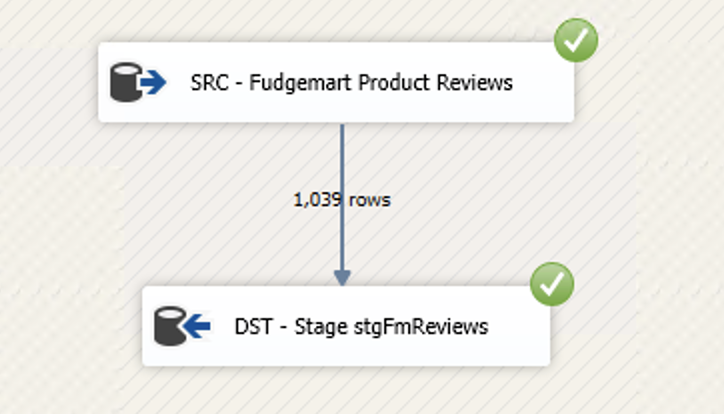
## Stage FudgeMart Product Reviews

The package that staged the FudgeMart Product reviews contained one SQL Task and one data flow task. The SQL task only truncated the stgFmReviews table, and the only extracted the reviews from the FudgeMart database and inserted the reviews into the stgFmReviews table. There was no staging of the FudgeMart Products or the FudgeMart Customers into the stage database because those dimensions would be already staged and loaded from the Total Sales ETL process.

**Control Flow FudgeMart Product Reviews**

****

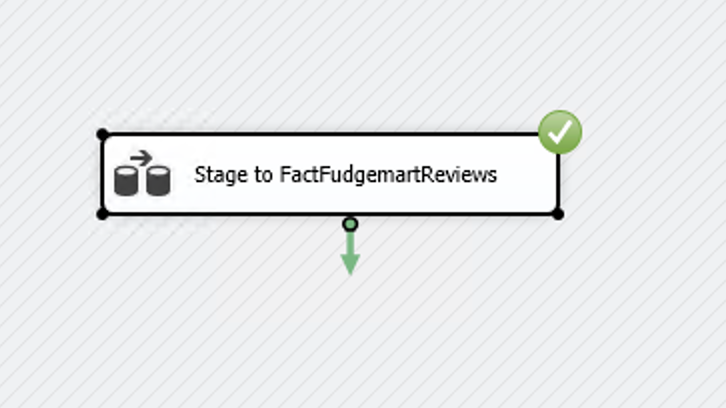
**Data Flow Stage FudgeMart Product Reviews**

****

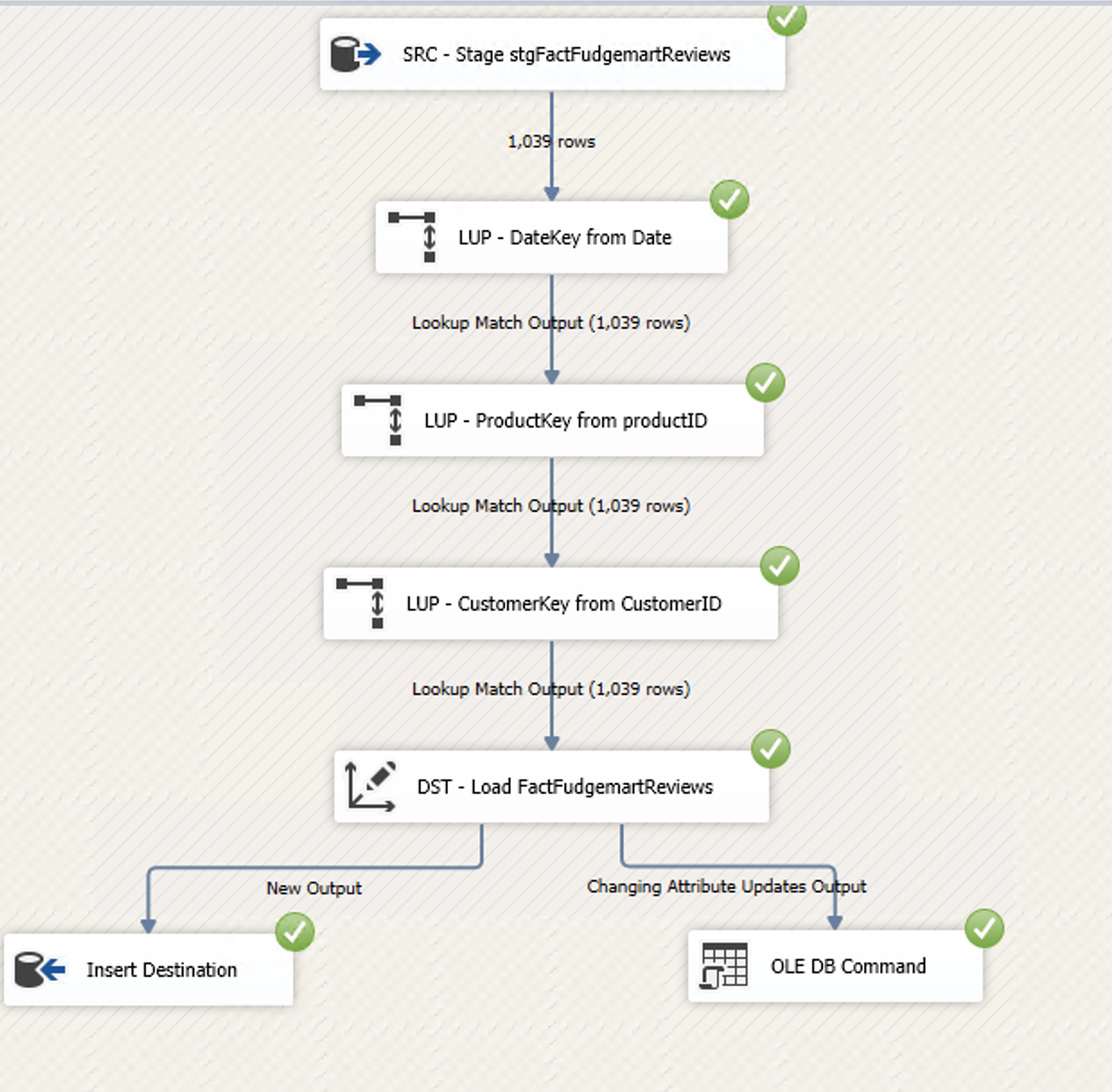
## Data Warehouse Fact FudgeMart Reviews

The last package created was designed to insert the FudgeMart product reviews from the staging database into the data warehouse. The package only required one data flow task inserting the product reviews into the data warehouse because the Total Sales Fact is designed to run first, and will already insert all of the Customers and Products into the Data Warehouse. There are not any derived columns, or casting of data types. Before inserting the review into the data warehouse, the package will lookup the DateKey from the review\_date column, the ProductKey from the product id column where the subsidiary is equal to “FudgeMart”, and the CustomerKey from the customer id column where the subsidiary is equal to “FudgeMart”. The product review is then inserted into the data warehouse as a Type 1 Slowly Changing Dimension with the customer key and the product key as the business keys.

**Control Flow for FudgeMart Product Reviews**

****

**Data Flow for FudgeMart Product Reviews into FactFudgemartReviews**

****

## Combining all of the Packages

After all of the packages staging the data and loading the data into the data warehouse had been created, we combined all of the packages together so we could stage and load all of the data in one step. First, the Date Dimension was imported into the data warehouse, followed by the staging of the data for total sales. After the total sales data had been staged, the package loading the FudgeMart sales and the package loading the FudgeFlix sales were run in parallel. This ensured that all of the conformed dimensions had been loaded into the data warehouse. After the loading the required data from the FudgeMart Sales into the data warehouse, the FudgeMart Customer Reviews were staged, and then loaded into the data warehouse. The reason that the loading of the FudgeMart customer reviews followed the loading of the Total sales is because we needed to make sure that all of the dimensions had been imported into the data warehouse so the customer and product keys could be properly looked up.

**Control Flow Package**

