

ETL PROCESS

Introduction:

We will be implementing data warehouse sales business line for Fudge Enterprises for both of their business that is Fudgeflix which has business of renting online DVD by mail and video on demand service and Fudgemart has an online retail business. For sales, we have identified the common dimension for our data warehouse. Following table describes about it.

Dimension and Fact table	Source table from Fudgemart	Source table from Fudgeflix
DimCustomer	fm_customers	ff_accounts, ff_zipcodes
DimProducts	ffm_products	ff_plans
FactSales	fm_Orders and fm_order_details	ff_account_billing
DimDate	date_dimension from externalsources database	date_dimension from externalsources database

Following are the staging tables that we will be using in staging database to store the data from sources:

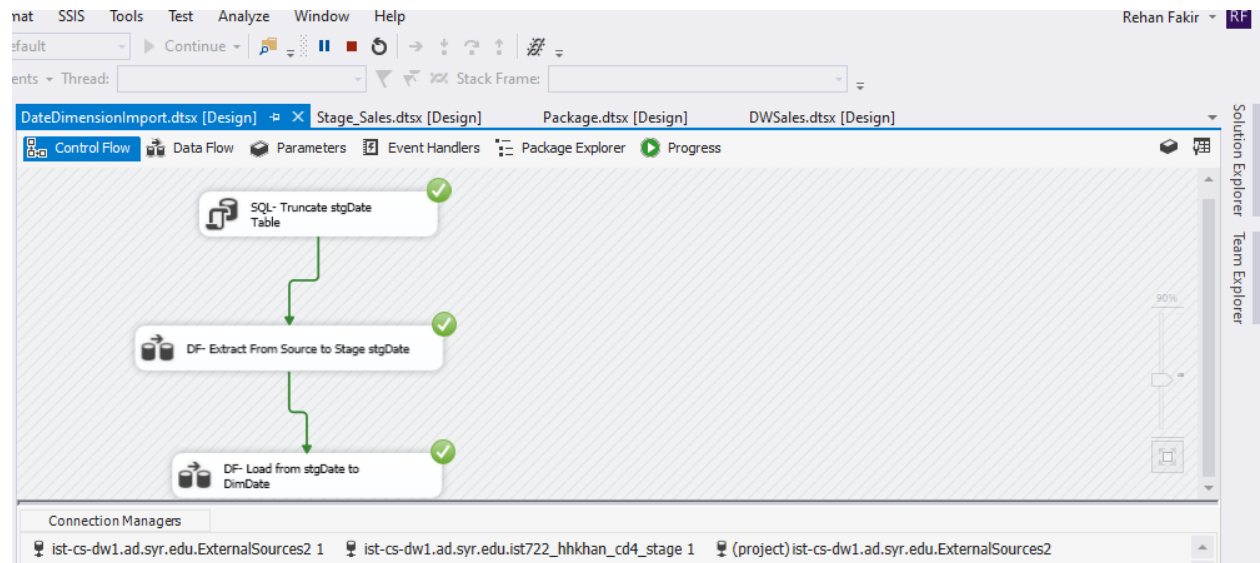
- stgDate
- stgFudgeflixaccounts
- stgFudgeFlixPlans
- stgFudgeFlixSales
- stgFudgeMartCustomers
- stgFudgeMartProducts
- stgFudgeMartSales

Using ETL platform SSIS, data warehouse will be created and loaded with the data. Once the data is loaded into the data warehouse, using SSAS tool, cube (MOLAP) will be created. The star schema for both the SSAS and SSMS is added at the end of this document. Using the Cubes and data warehouse, dashboards, pivot tables and charts are created to highlight the trends, anomalies in the sales business line of the company.

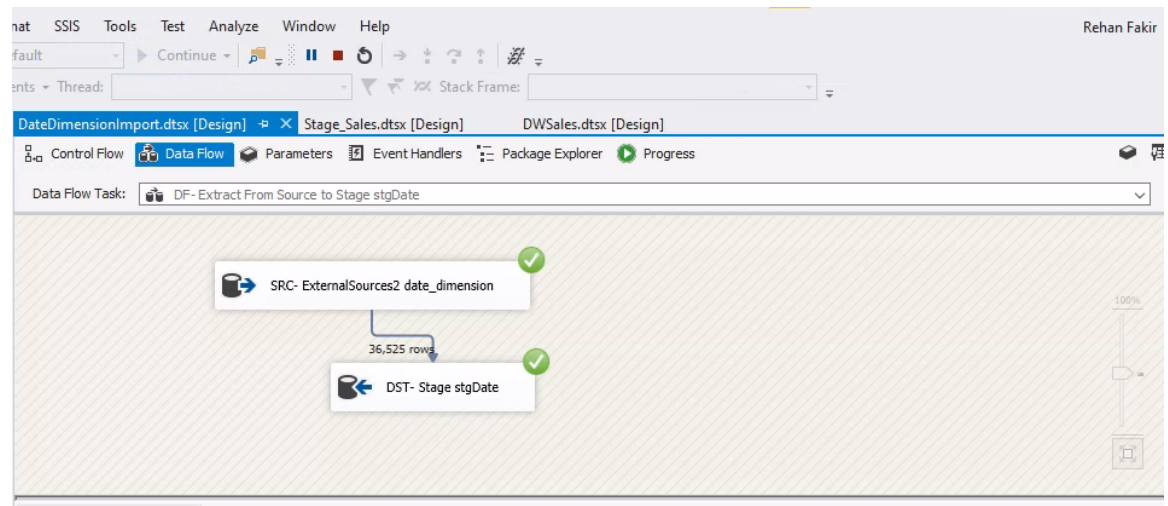
ETL PROCESS

1) Date stage and DimDate

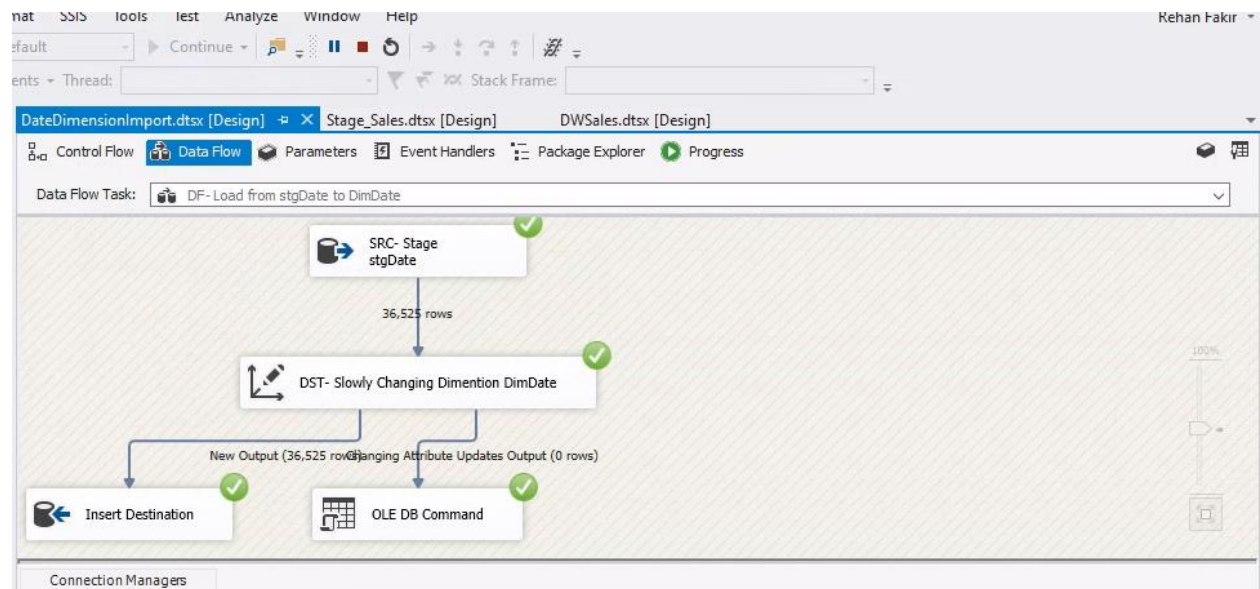
Date data is loaded from Externalsources database into our stage table stgDate. From Staging, date data is moved to DimDate dimension table in datawarehouse. While doing this, slowly changing dimension Type 1 is used to update the old record to new record when there is a change in the record.



DF –Extract From source to stgDate: Date from the Externalsources database date_dimension table is used as source and loaded into the created table stgDate.

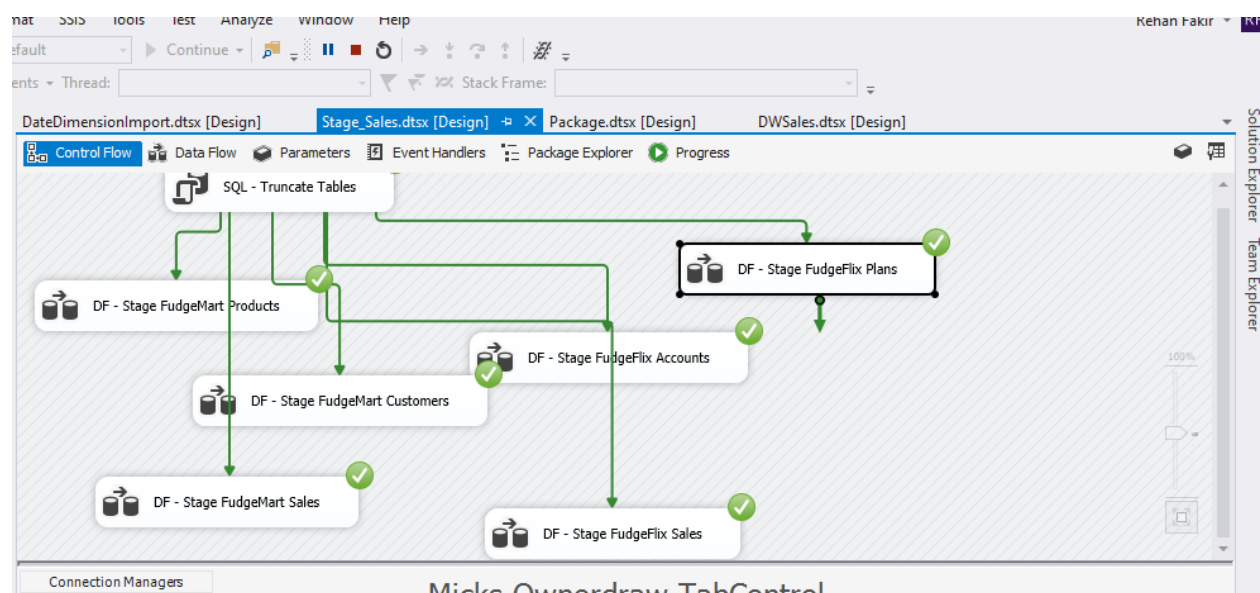


DF – Load from stgDate to DimeDate: Data from staging is loaded into dimension date in data warehouse using slowly changing dimension type 1.



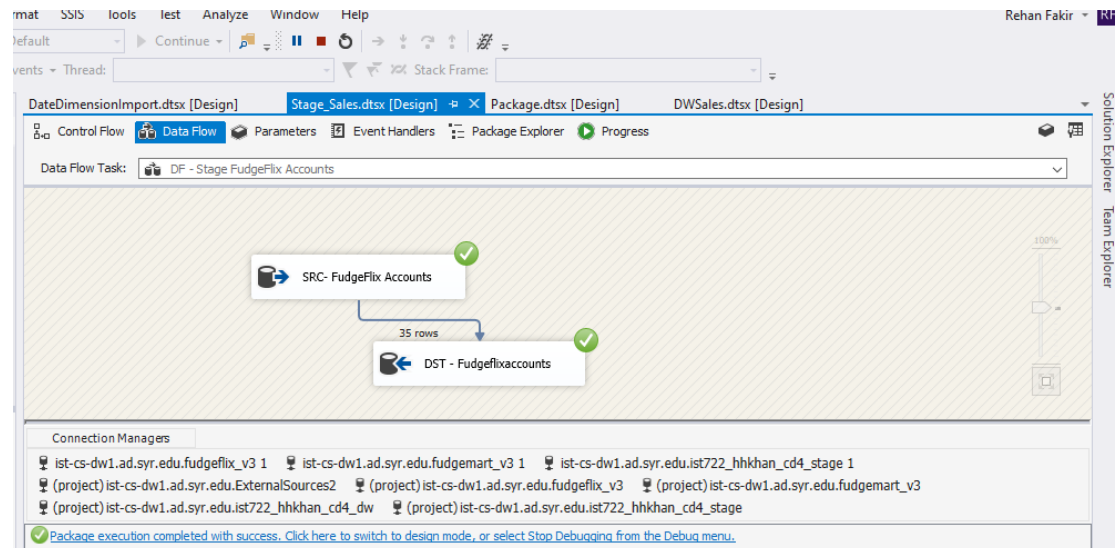
2) Moving data from source to stage tables

Before moving data into data warehouse, data is stored in staging tables by loading into it from source.



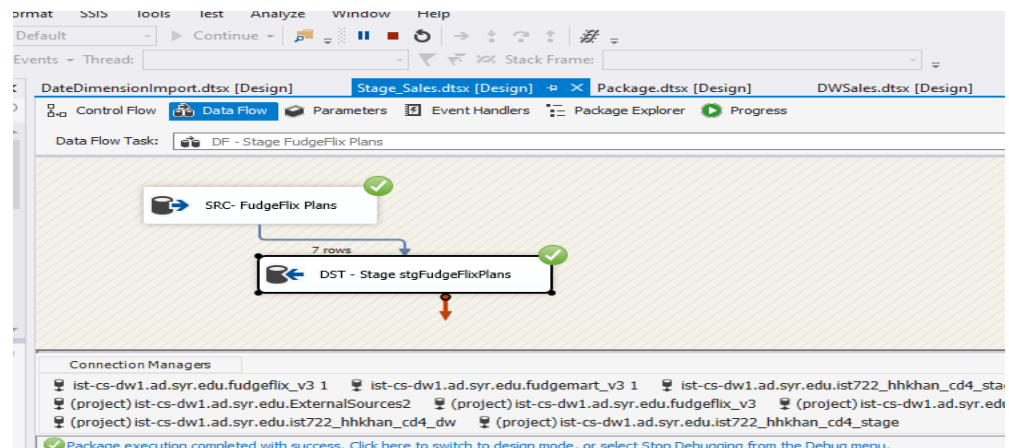
a) Stage stgFudgeflixaccounts:

The table is truncated in the first step so that every time the package is run, if the table is present then it is truncated and new data will be loaded in the next step. The staging table is created in DST – Fudgeflixaccounts. And then the data is mapped and moved into newly created staging table. A column Source having data Fudgeflix is created while creating the staging table.



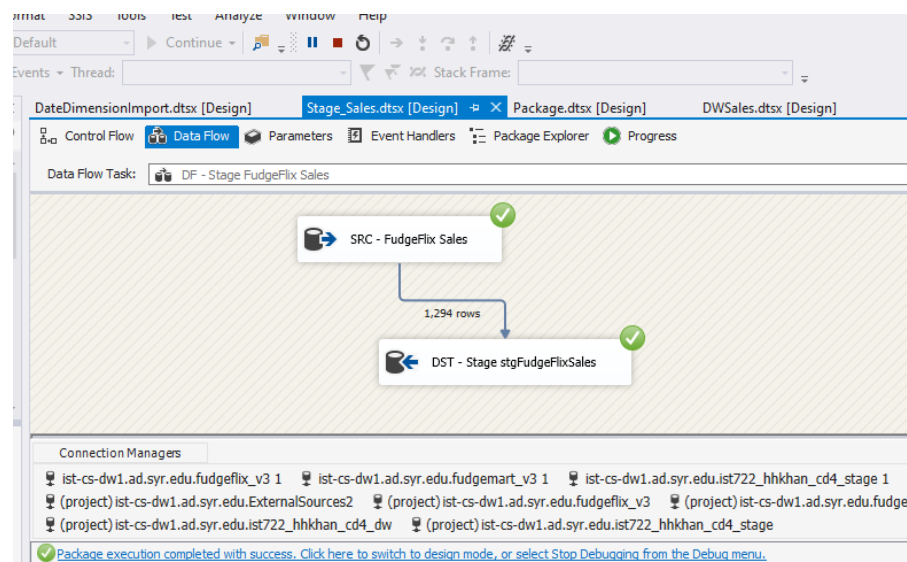
b) Stage stgFudgeFlixPlans:

The table is truncated in the first step so that every time the package is run, if the table is present then it is truncated and new data will be loaded in the next step. The staging table is created in DST – FudgeFlixPlanss. And then the data is mapped and moved into newly created staging table. A column Source having data Fudgeflix is created while creating the staging table.

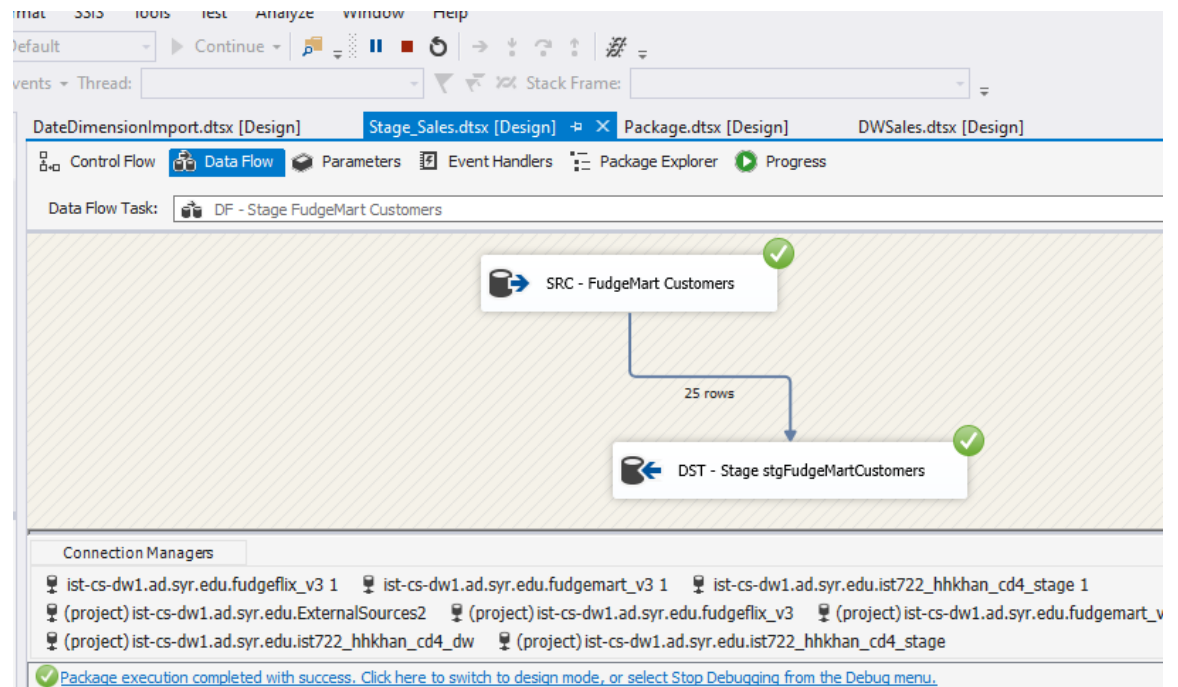


c) Stage stgFudgeFlixSales:

The table is truncated in the first step so that every time the package is run, if the table is present then it is truncated and new data will be loaded in the next step. The staging table is created in DST – Fudgeflixsales. And then the data is mapped and moved into newly created staging table. A column Source having data Fudgeflix is created while creating the staging table. Quantity=1 is also created as the customer only subscribes 1 type of plan at a time. For example, customer will opt for 1 plan A and not 2.

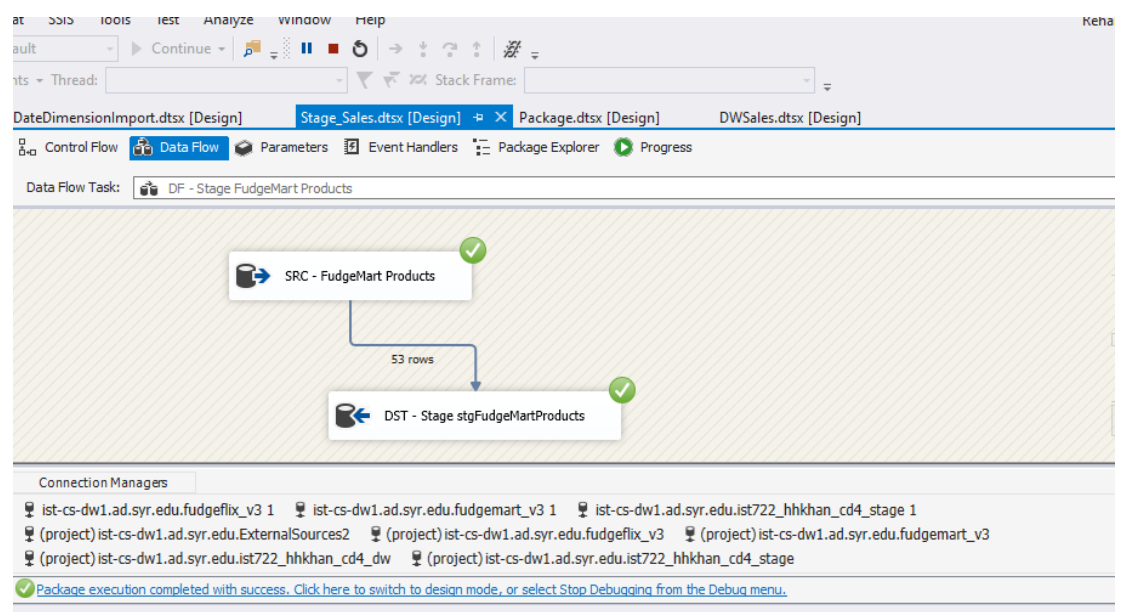
**d) Stage stgFudgeMartCustomers:**

The table is truncated in the first step so that every time the package is run, if the table is present then it is truncated and new data will be loaded in the next step. The staging table is created in DST – FudgeMartCustomers. And then the data is mapped and moved into newly created staging table. A column Source having data Fudgemart is created while creating the staging table.



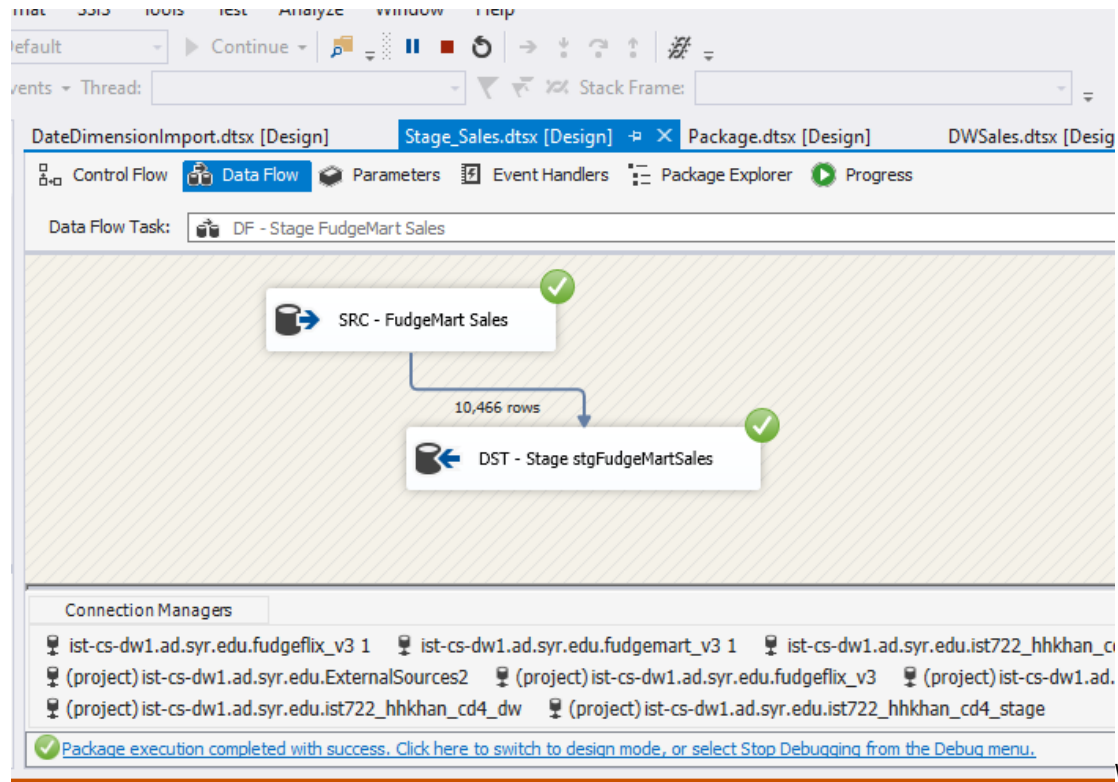
e) Stage stgFudgeMartProducts:

The table is truncated in the first step so that every time the package is run, if the table is present then it is truncated and new data will be loaded in the next step. The staging table is created in DST – FudgeMartProducts. And then the data is mapped and moved into newly created staging table. A column Source having data Fudgemart is created while creating the staging table.

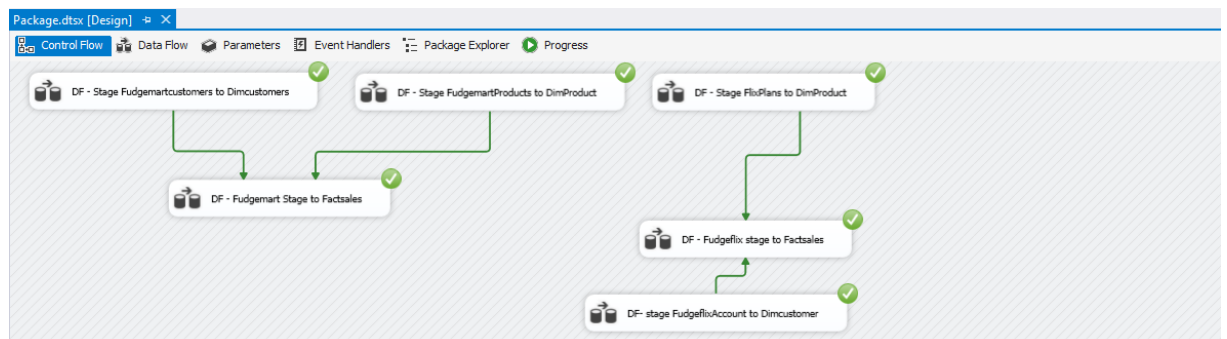


f) Stage stgFudgeMartSales:

The table is truncated in the first step so that every time the package is run, if the table is present then it is truncated and new data will be loaded in the next step. The staging table is created in DST – FudgeMartSales. And then the data is mapped and moved into newly created staging table. A column Source having data Fudgemart is created while creating the staging table.

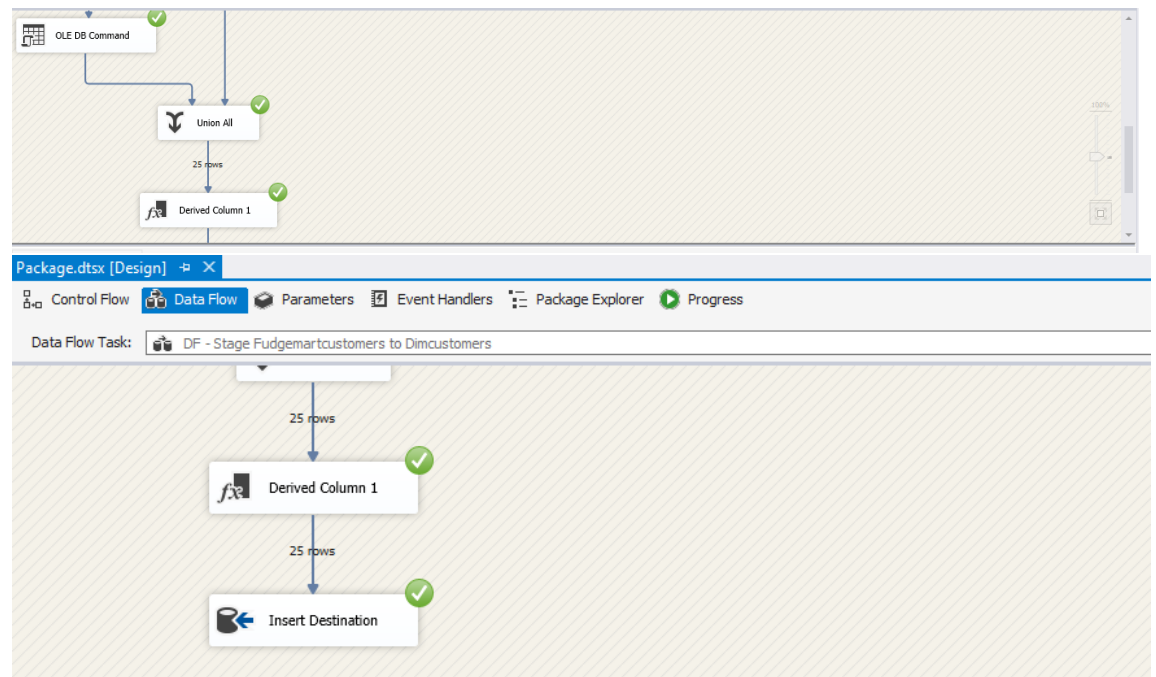
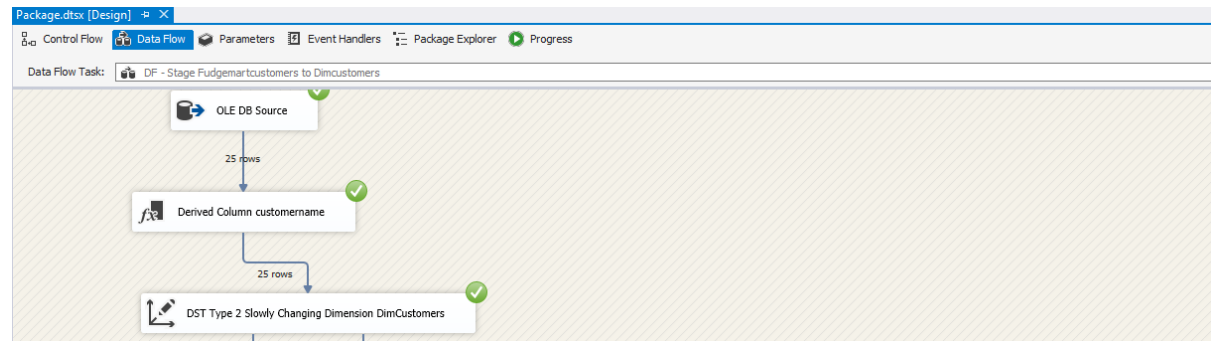
**3) Moving Data from Stage table to Data warehouse:**

Now, we have moved data from stage tables to dimension and fact tables. Following is the data flow diagram of the same.



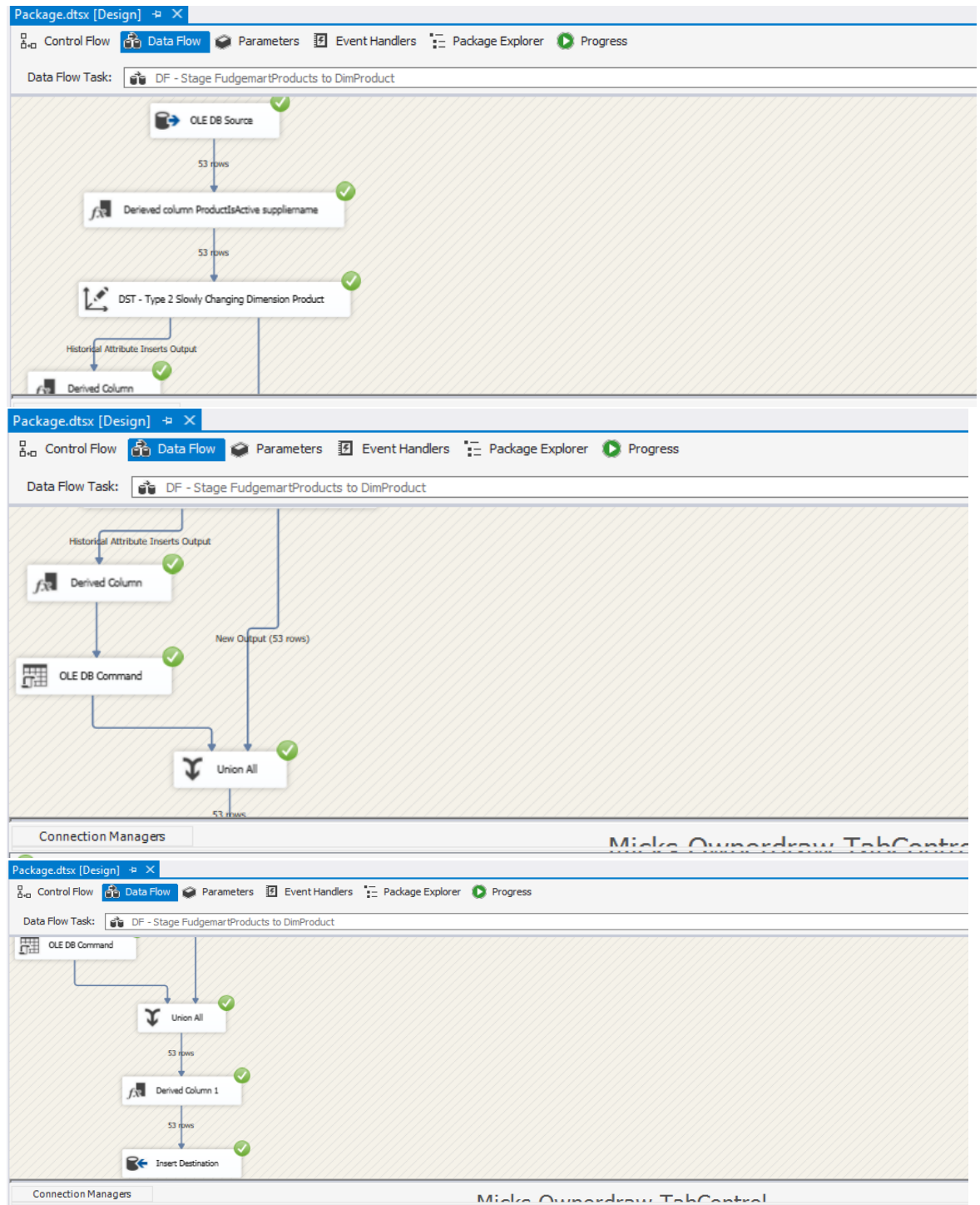
a) Stage FudgeMartCustomers to DimCustomers:

We identified stgFudgeMartCustomers as source table and derived column customername from firstname and lastname column. We are using slowly changing dimension type 2 and identified customerId as business key. And rest all attributes like customername, customer_mail, customer_city and customer_state as input columns and as historical attributes from stgFudgeMartCustomers table.

**b) Stage stgFudgeMartProducts to DimProducts:**

We identified stgFudgeMartProducts as source table and derived column suppliername from vendor_name and converted Product_is_active attribute to Yes or No from bit datatype. We are using slowly changing dimension type 2 and

identifies product_Id as business key. And rest all attributes like product_name, product_Is_Active, suppliername and department_name as input columns and as historical attributes from stgFudgeMartProduct table.

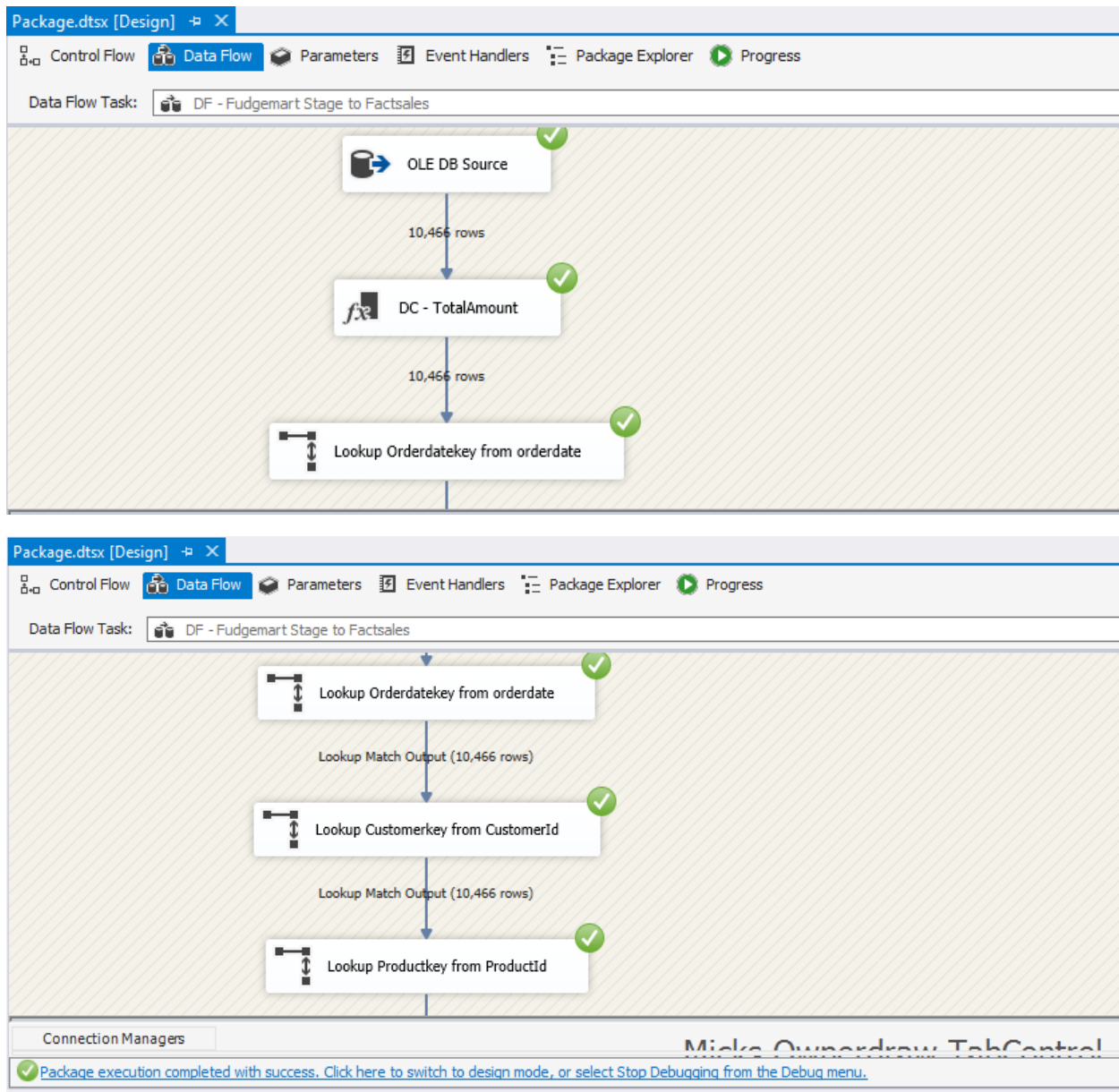


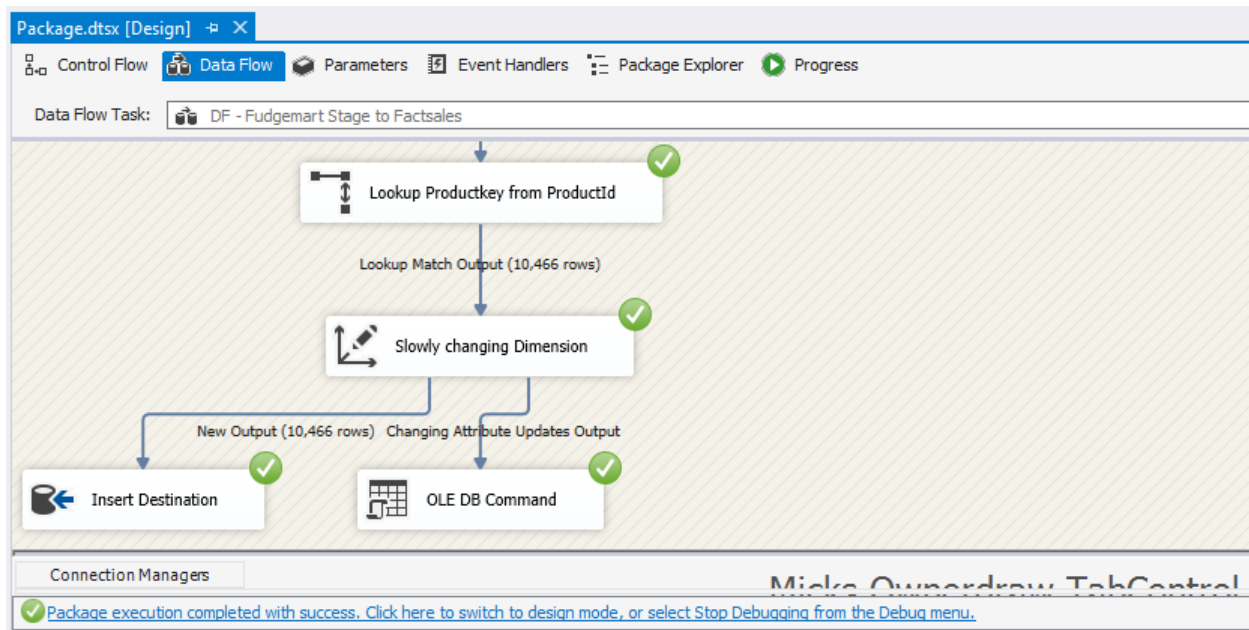
c) Stage stgFudgeMartSales to FactSales

We identified stgFudgeMartSales as source table and derived column TotalAmount from Quantity and UnitPrice attributes of the table with formula

$$\text{TotalAmount} = \text{UnitPrice} * \text{Quantity}$$

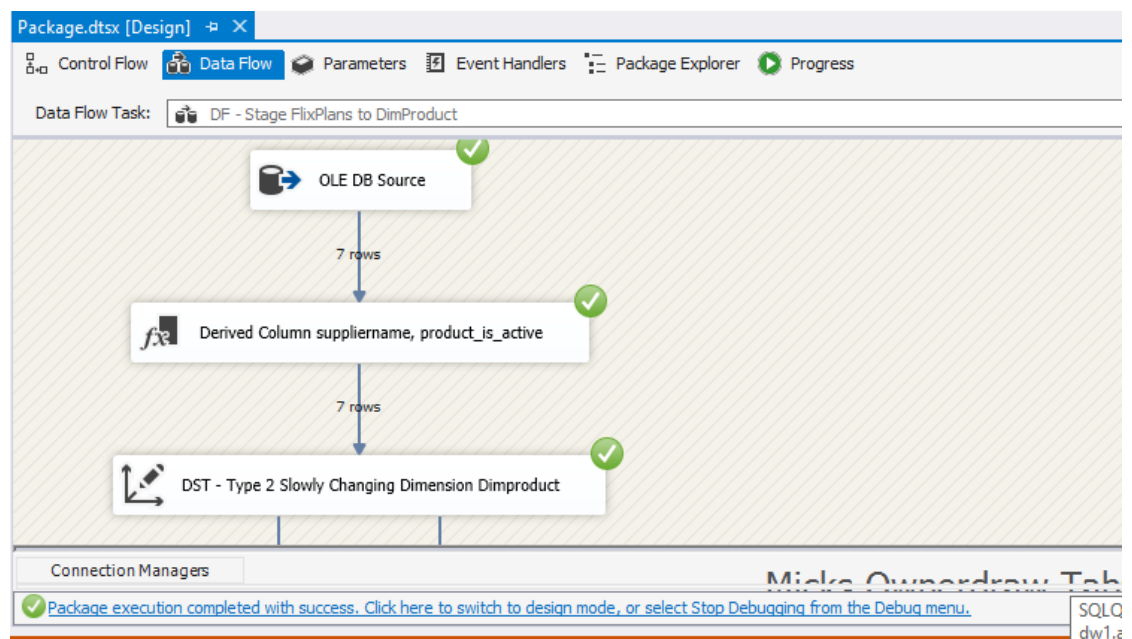
Using lookup, found OrderdateKey, ProductKey and CustomerKey from DimDate, DimProduct and DimCustomer dimensions respectively. Then using Slowly Changing Dimension, identified OrderID and Productkey as Business key a rest all attributes were kept as Changing attributes. Finally data was inserted into FactSales.

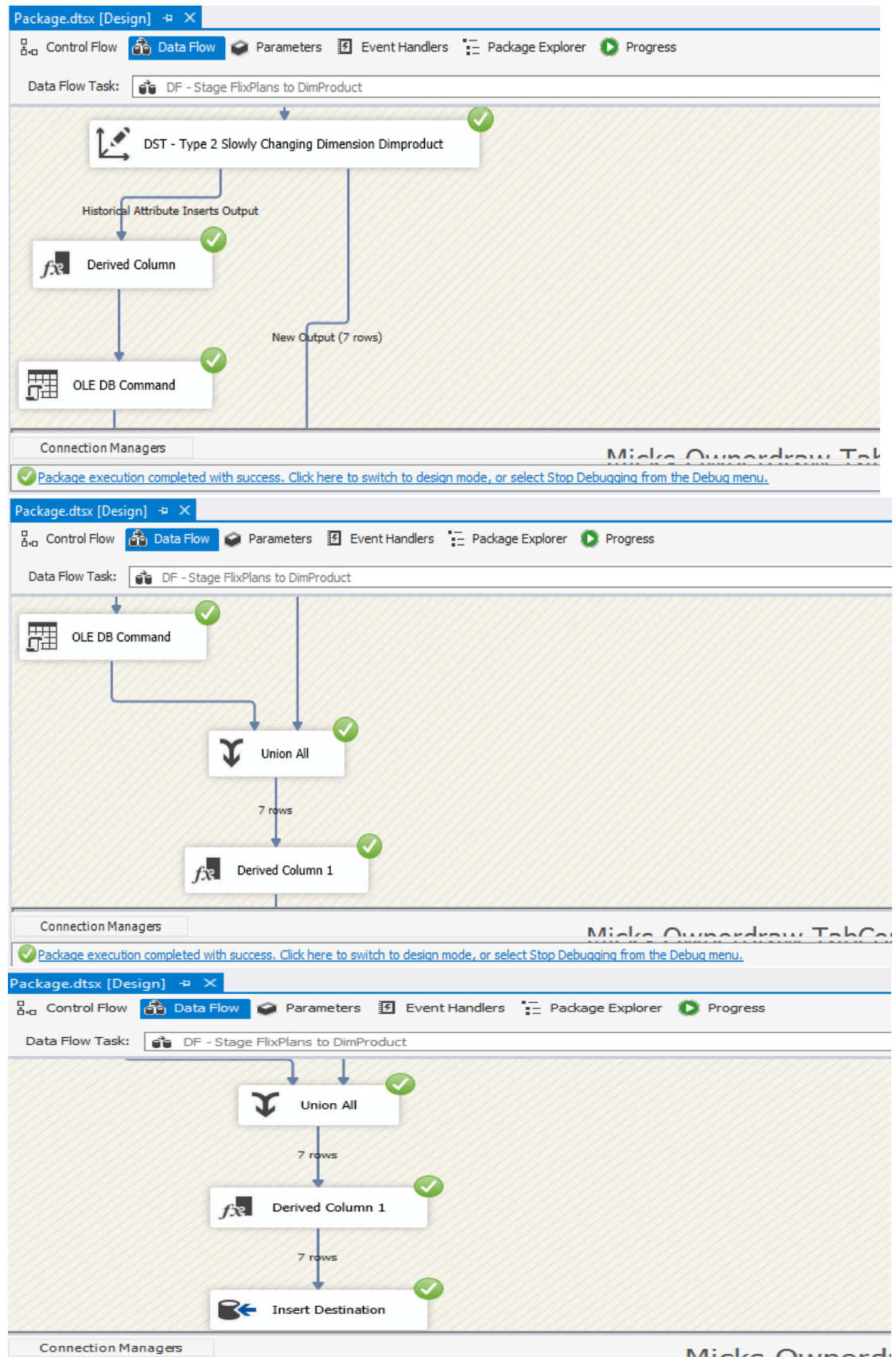




d) Stage stgFudgeFlixPlans to DimProducts:

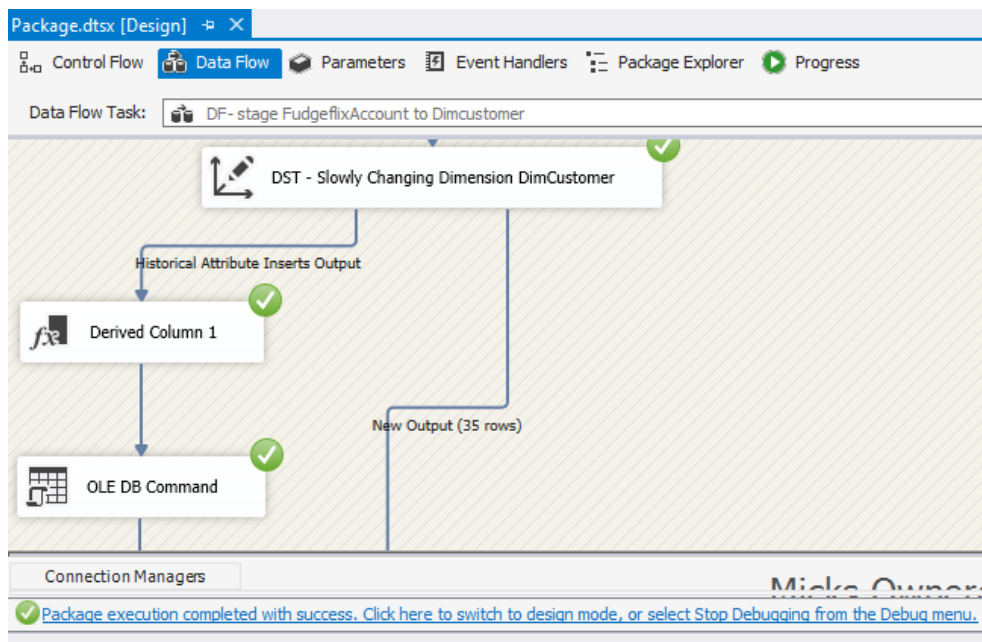
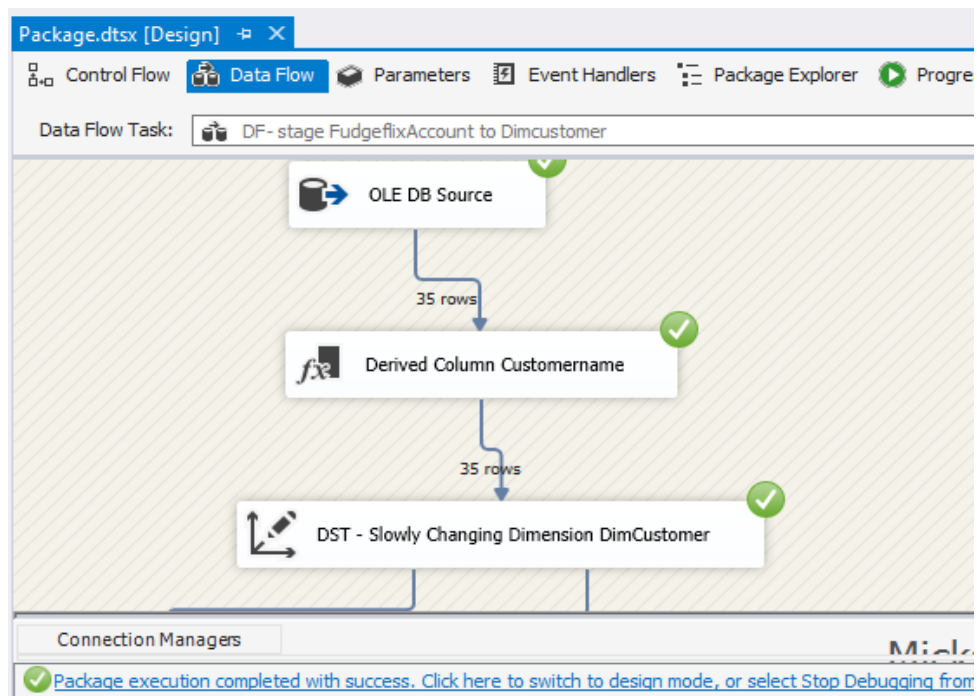
We identified stgFudgeflixPlans as source table and derived column Product_Is_Active column by converting plan_current bit attribute of ff_plans to cilumn consisting of Y or N. We are using slowly changing dimension type 2 and identified plan_id as business key. And rest all attributes are input columns and set as historical attributes from stgFudgeflixAccounts table.

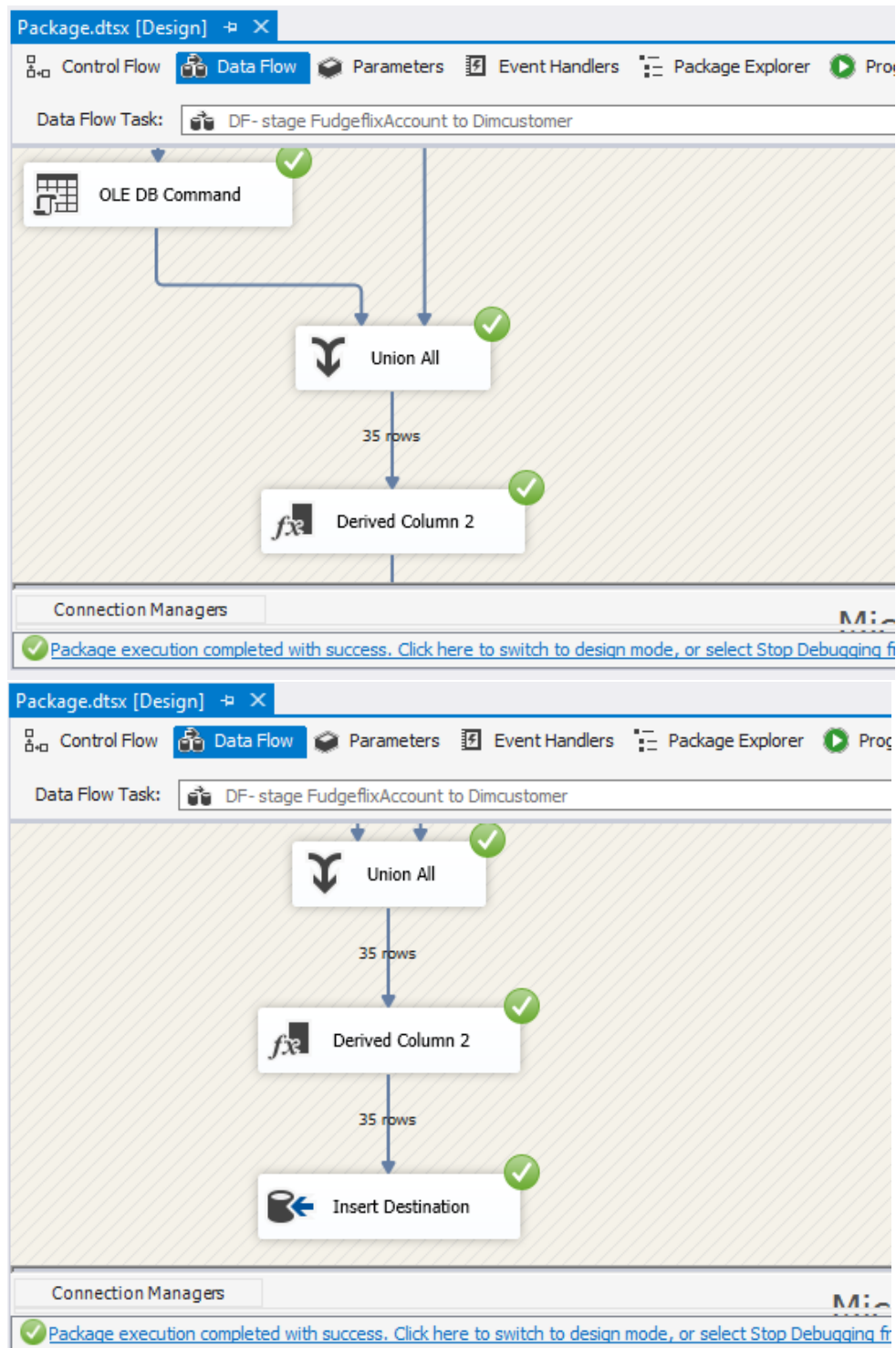




e) Stage StgFudgeFlixAccounts to DimCustomers:

We identified stgFudgeflixAccounts as source table and derived column account_firstname and account_lastname column. We are using slowly changing dimension type 2 and identified account_id as business key. And rest all attributes are input columns and set as historical attributes from stgFudgeflixAccounts table.



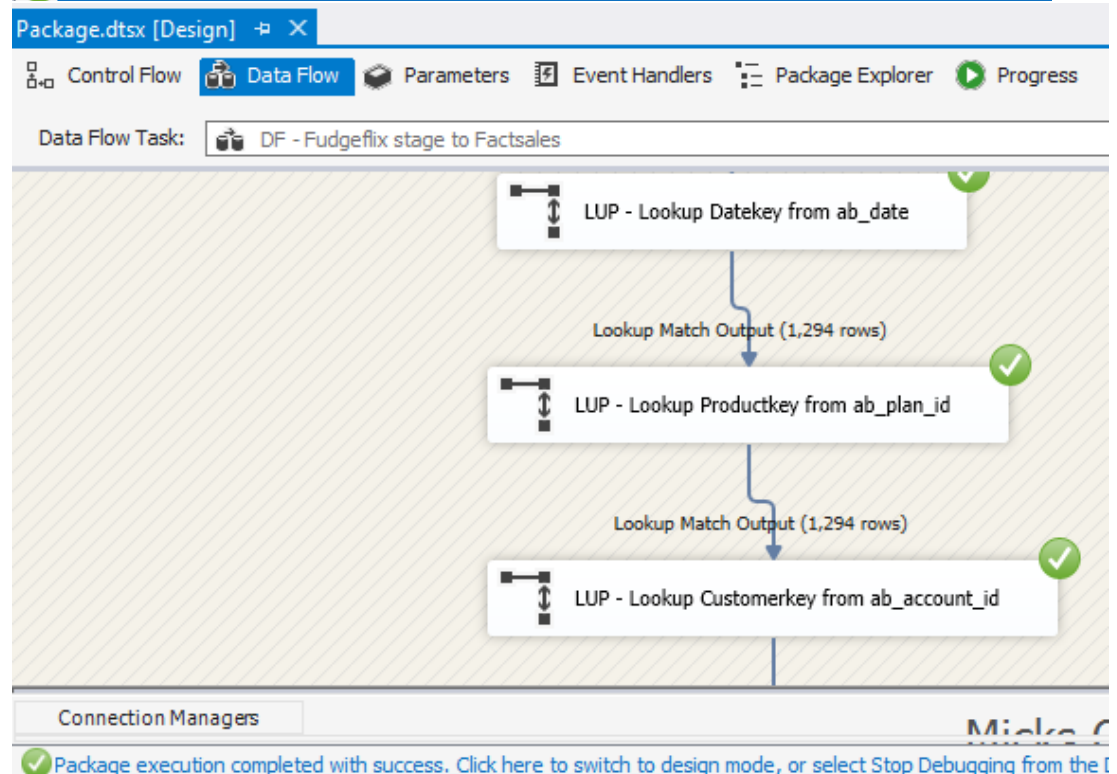
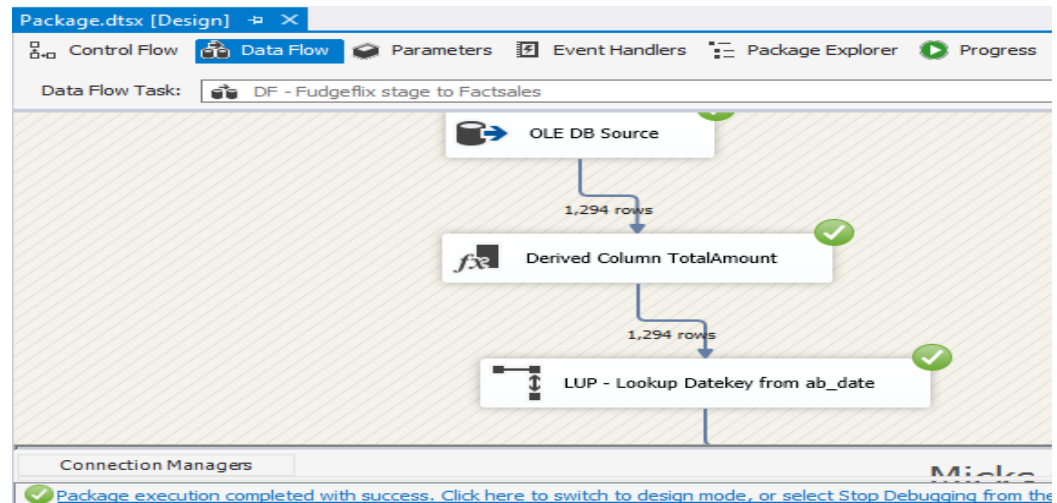


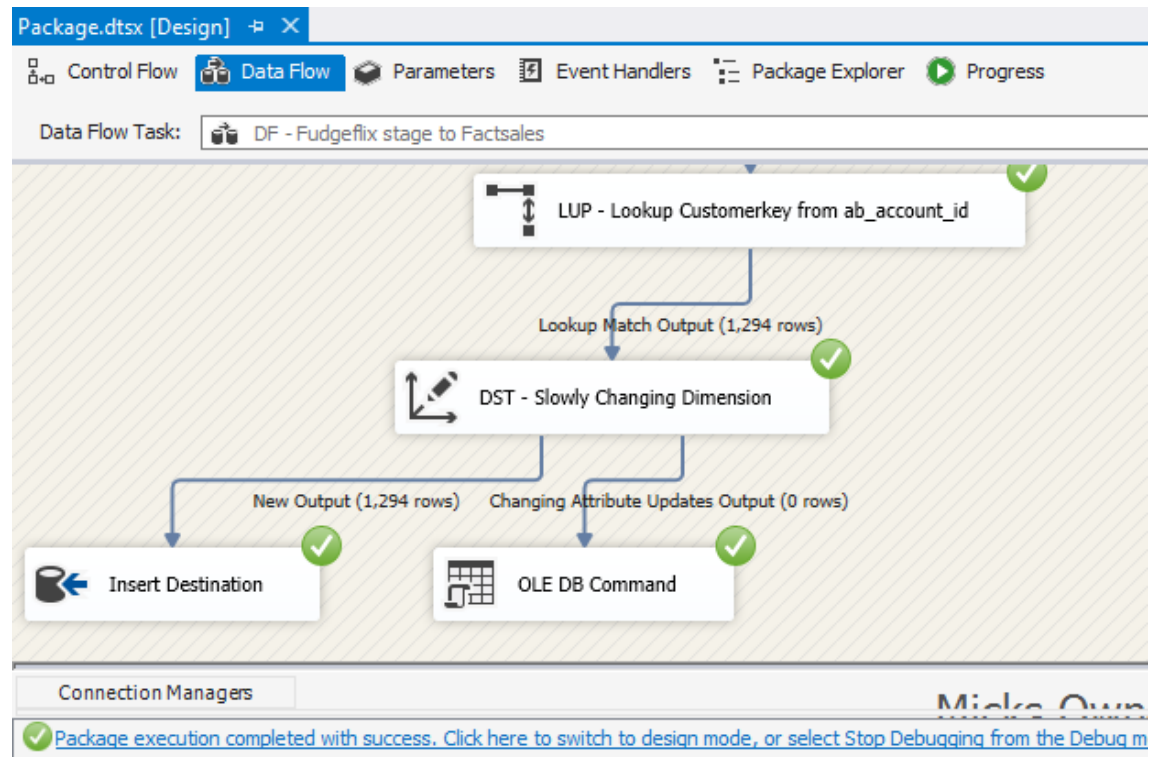
f) Stage stgFudgeFlixSales to FactSales:

Used stgFudgeFlixSales table as source and used derieved column to derive TotalAmount column using formula:

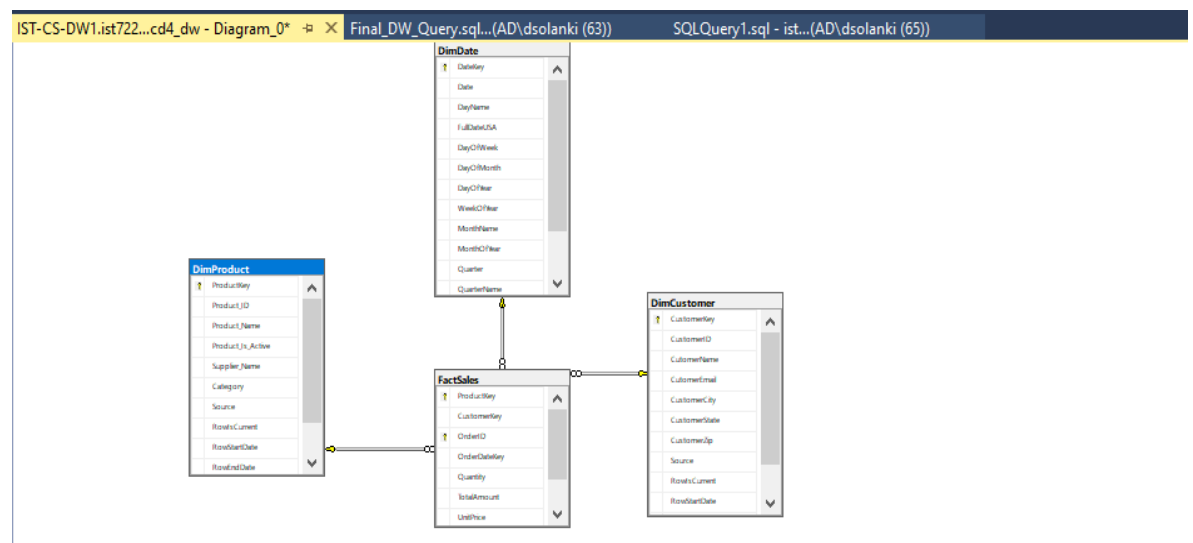
$$\text{TotalAmount} = \text{Plan_price} * \text{Quantity}(1)$$

Used lookup to find orderdatekey, productkey and customerkey from Dimdate, DimProduct and DimCustomer using Date, ProductID and CustomerID respectively. For slowly changing dimension, the business keys identified are OrderID and Productkey. Rest all the attributes were considered as not a business key and set to changing value.





Star schema in SSIS:



Star schema in MOLAP (SSAS):