DSS – SECOND ASSIGNMENT

Domenico Sacco
Thiago da Silva Alves

REPORT CONTENTS

1.DATA WAREHOUSE STRUCTURAL CHANGES 2
1.1 Product 2
1.2 SalesTerritory 2
1.3 City, StateProvince and Country 2
1.4 Dimension used 2
2.ETL PROCESS 3
2.1 Category table 3
2.2 SubCategory Table 3
2.3 Product Table 4
2.4 DiscountPolicy Table 4
2.6 SalesOrderHeader Table 6
2.7 Time Table 6
2.8 SalesPerson Table 7
2.9 Country 8
2.10 StateProvince 8
2.11 City 8
2.12 Territory 9
2.13 SalesReason table 9
2.14 The FactTable 9
3.IMPLEMENTATION OF SLOW CHANGE DIMENSIONS 11
3.1 Type chosen 11
3.2 Implementation of the slow changing dimension 11
4.CONCLUSIONS 11

1.DATA WAREHOUSE STRUCTURAL CHANGES

1.1 Product

Product and ProductModel are still aggregated but the Category and Subcategory are divided, this modification was done to preserve aggregation power, making the classification by Subcategory and Category easier.

1.2 SalesTerritory

Instead of having SalesTerritory and Country aggregated as a subdimension of SalesInfo, SalesTerritory became an independent dimension and Country its subdimension.

1.3 City, StateProvince and Country

Those tables were added to provide the location of Customers and Stores

1.4 Dimension used

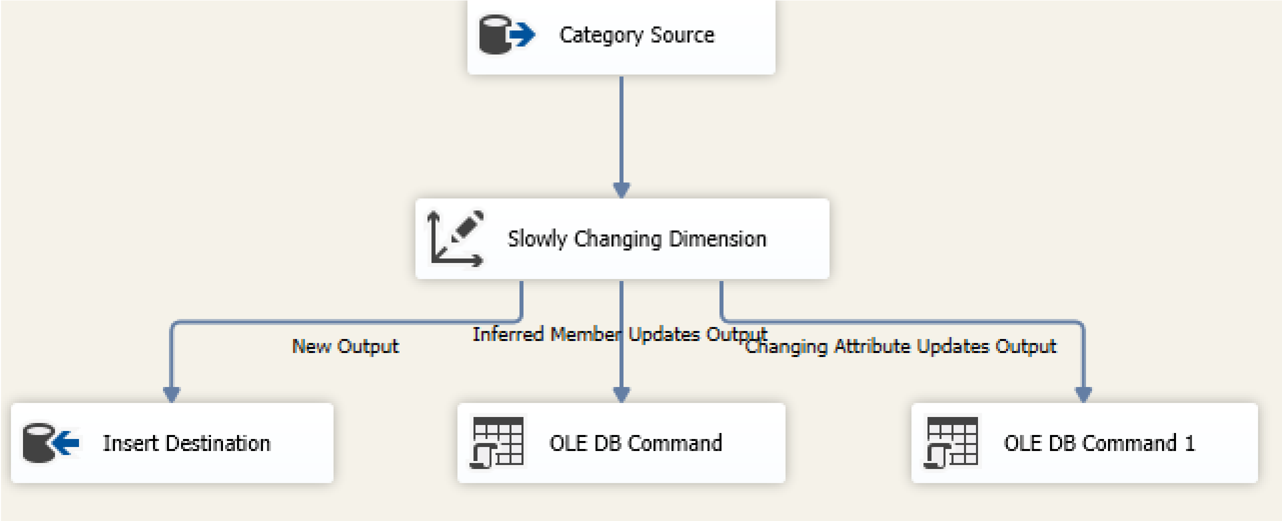
To summarize, those are the dimensions used in the data warehouse:

- **DimTime**
- **DimSalesPerson**
- **DimCustomer (City, StateProvince and Country as subdimensions)**
- **DimDiscount**
- **DimStore (City, StateProvince and Country as subdimensions)**
- **DimReason**
- **DimProduct (Subcategory and Category as subdimensions)**
- **DimSalesInfo**
- **DimTerritory (Country as subdimension)**

2.ETL PROCESS

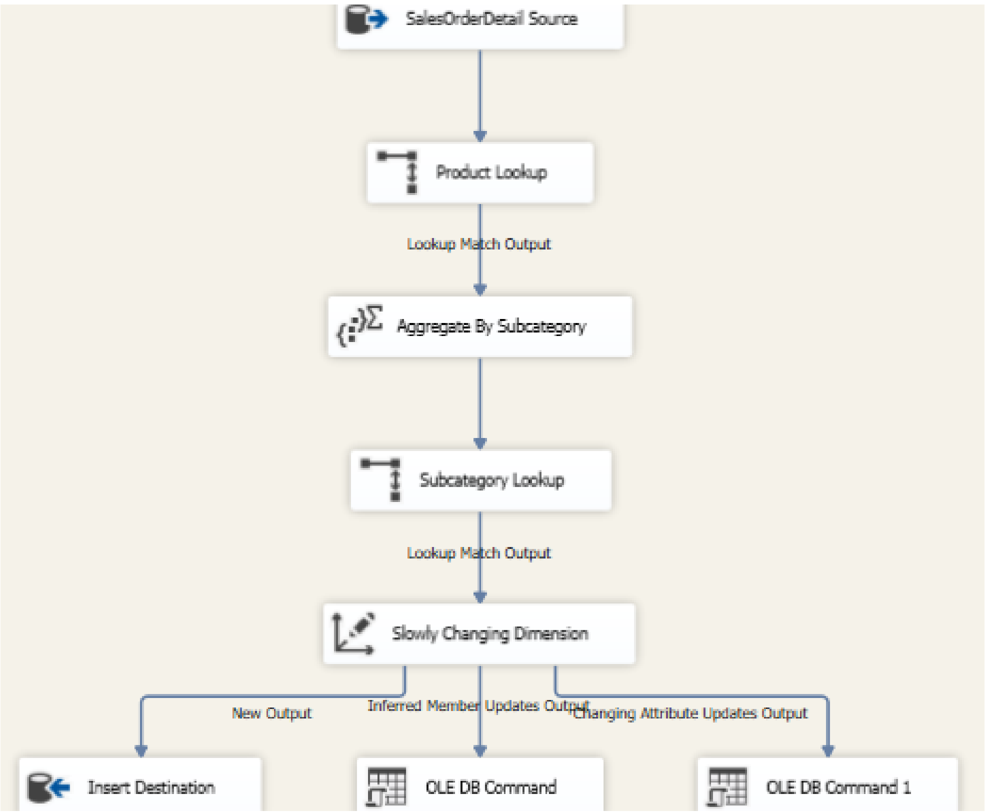
2.1 Category table

In this case the ETL process was simply to map ProductCategoryID and Name from AdventureWorks to the data warehouse.


Category Source
Slowly Changing Dimension
New Output
Inferred Member Updates Output
Changing Attribute Updates Output
Insert Destination
OLE DB Command
OLE DB Command 1

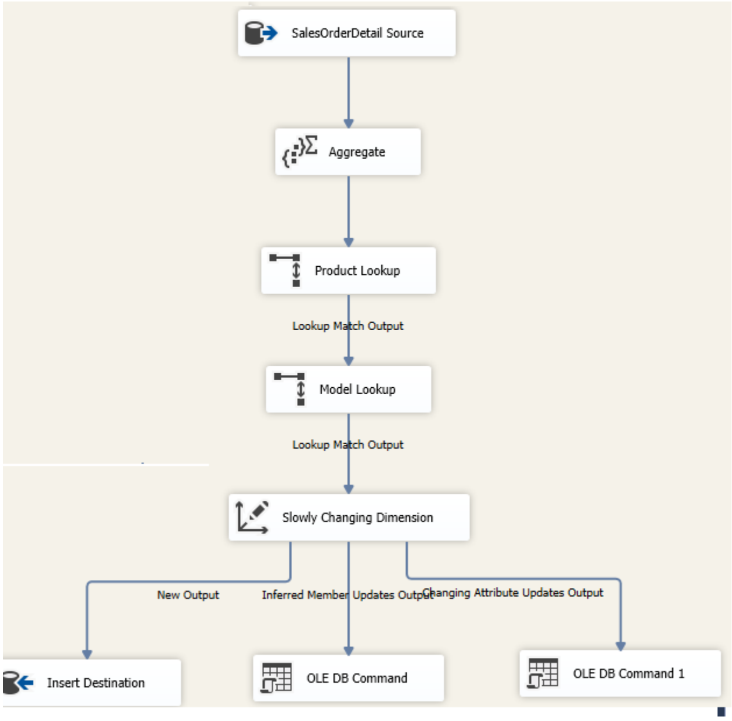
2.2 SubCategory Table

For this table SalesOrderDetail is the source and was necessary to use a lookup with the product table, an aggregation by subcategory and again a lookup with the subcategory table. SalesOrderDetail is used as source because in this way all the products that are sold at least once are considered.



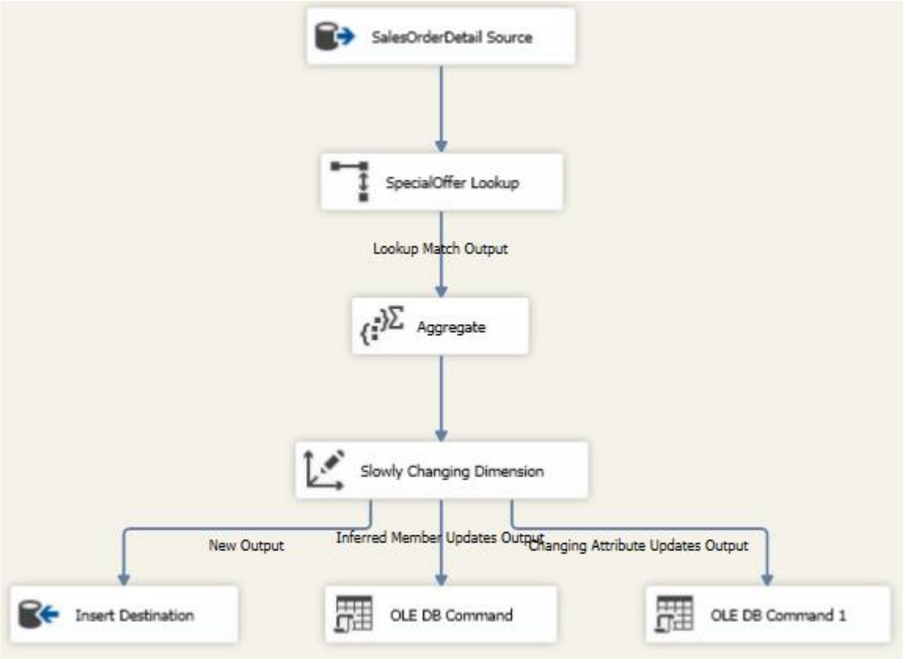
2.3 Product Table

In the actual data warehouse model, product also has its model features, for this table the ETL process consisted in extracting the data from SalesOrderDetail, aggregate them by product and lookup with product and model.



2.4 DiscountPolicy Table

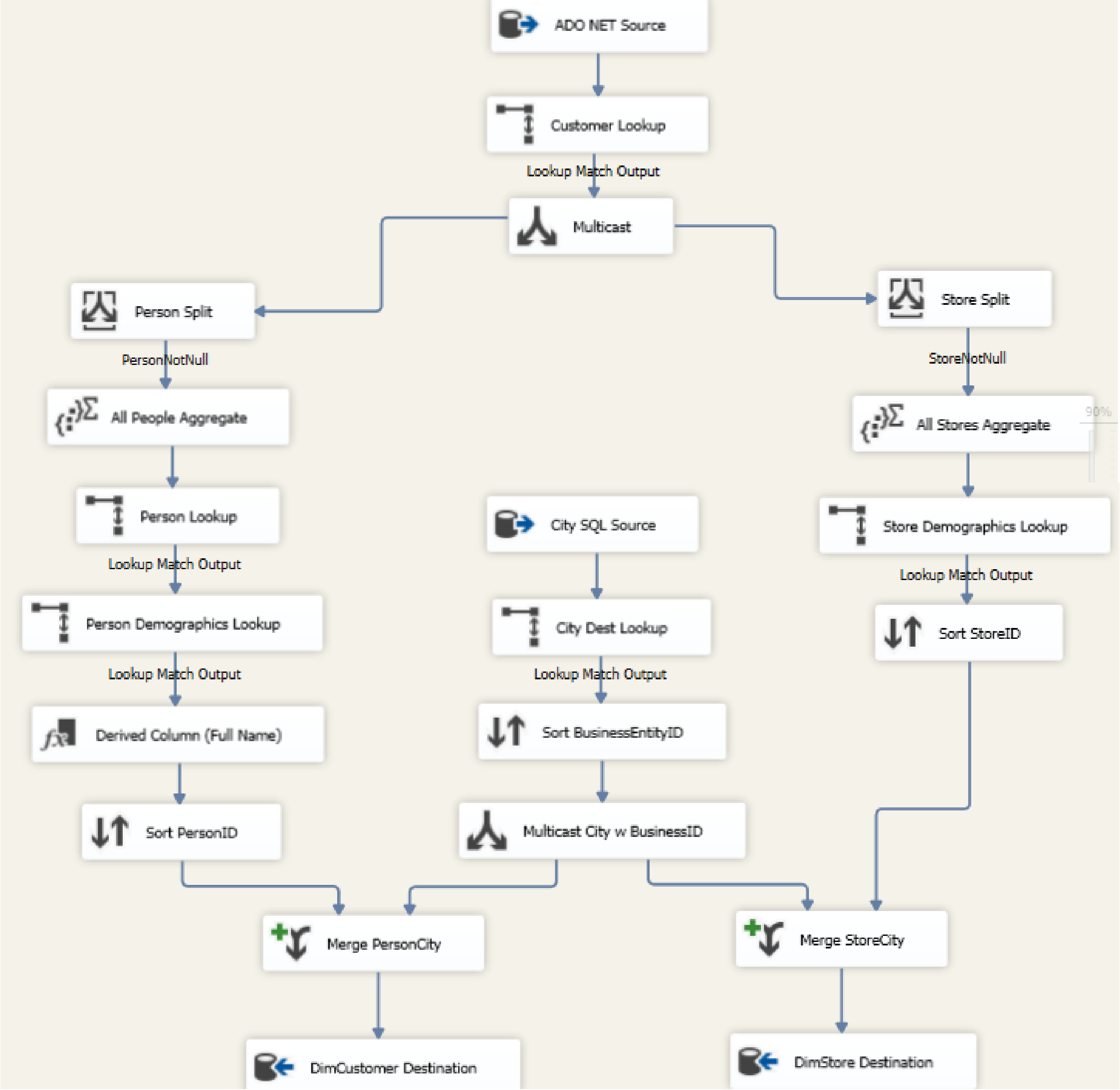
To consider products sold and discounts applied, the table SalesOrderDetail is the source, a lookup with SpecialOffer table is done, an aggregation by SpecialofferID, Description, DiscountPct, Type and Category follows.



2.5 Customer Table

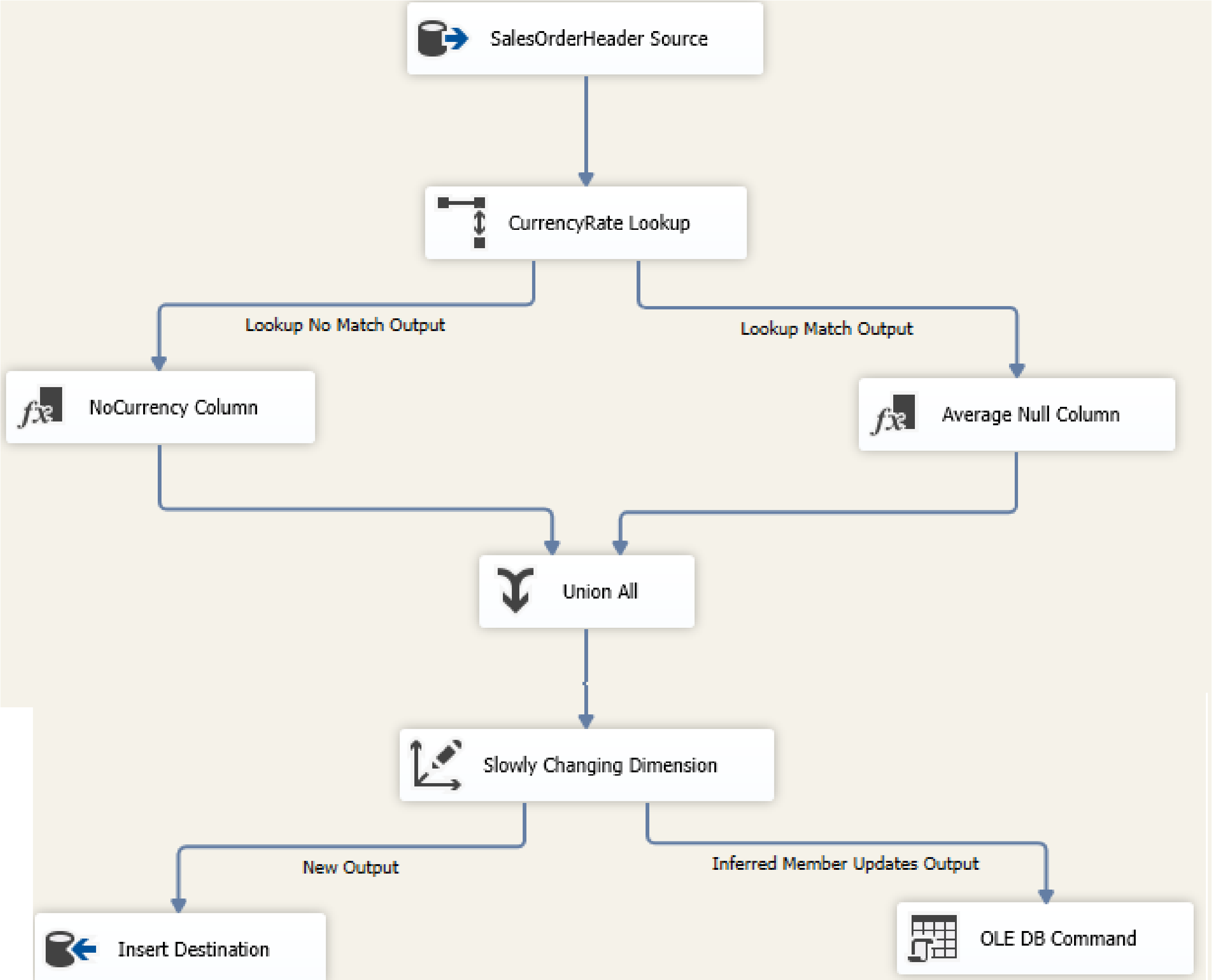
SalesOrderHeader was considered as source to consider only customers that already made an order (customers that never made an order do not have any useful information, beside the name, all other useful fields are NULL). A lookup with customer was done, after a multicast to direct the two outputs to two different conditional slits, the first has “PersonID is not null” as condition, the second “StoreID is not null” as condition. After each condition follows the aggregation by PersonID and by Store ID respectively. To the store branch follows a lookup to the Store table. The Person branch has two lookups, the first with Person and the second with Person Demographics, a derivate column to place the concatenation of first, middle and last name was used.

A merge with city table was performed to get the CityID from table city as source.



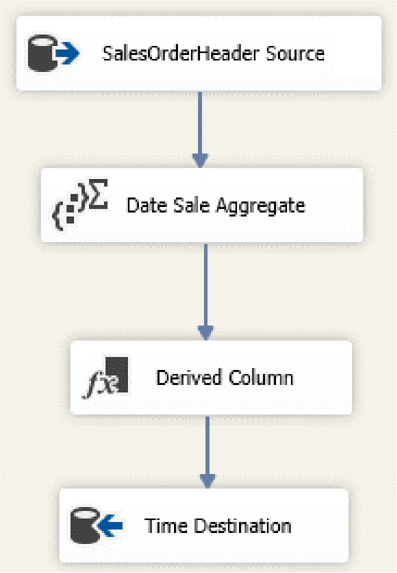
2.6 SalesOrderHeader Table

SalesOrderHeader was considered as source, but there was an important problem to solve: convert in the default currency (USD) all monetary values, therefore a lookup to the CurrencyRate table was performed. Two derivative columns are used, the first to set as one the average rate in case the currency is already USD, the second is used in case there is still an association with the currency table, but no information on how to convert the monetary value is provided.



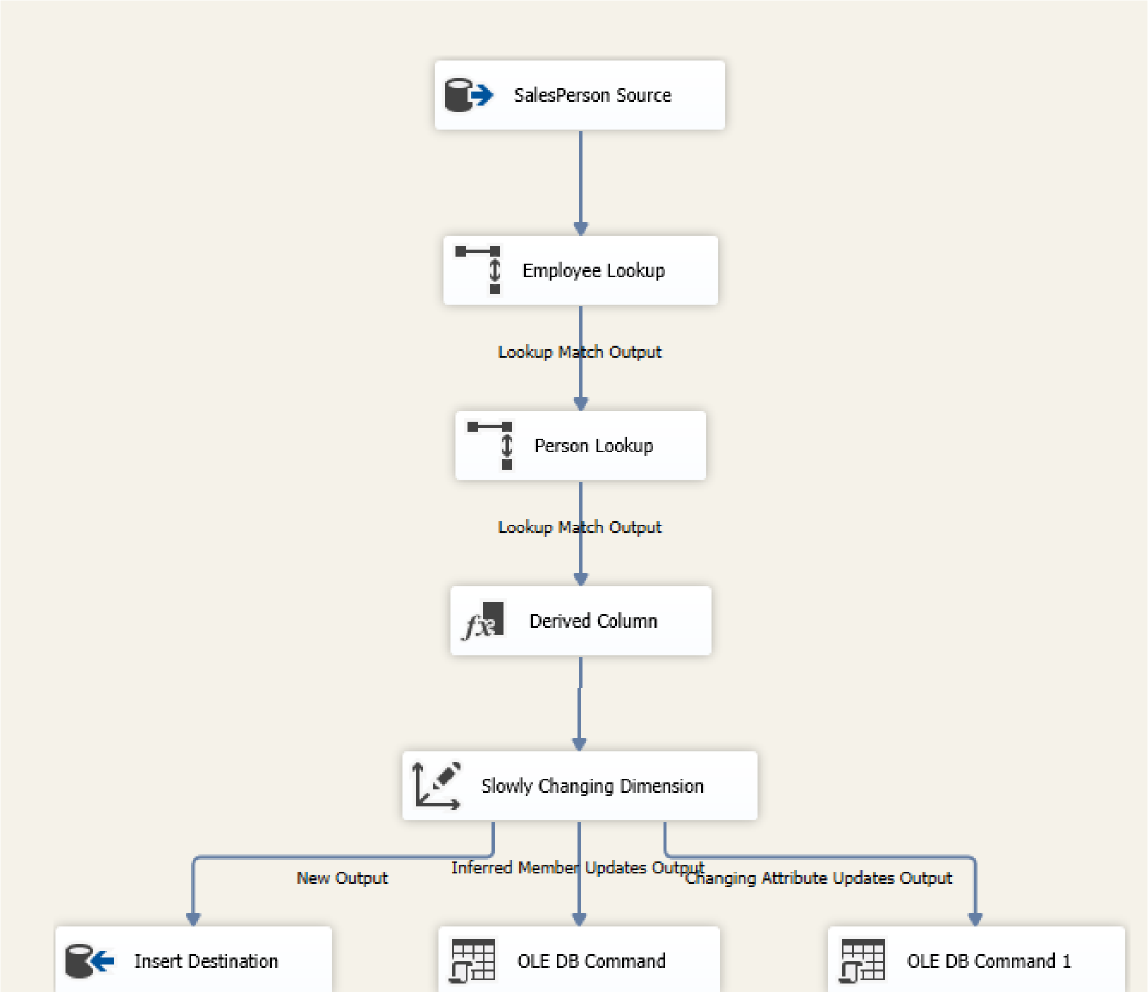
2.7 Time Table

SalesOrderHeader was used as source, an aggregation for sale date was performed and a derived column to extract year, month and day was used.



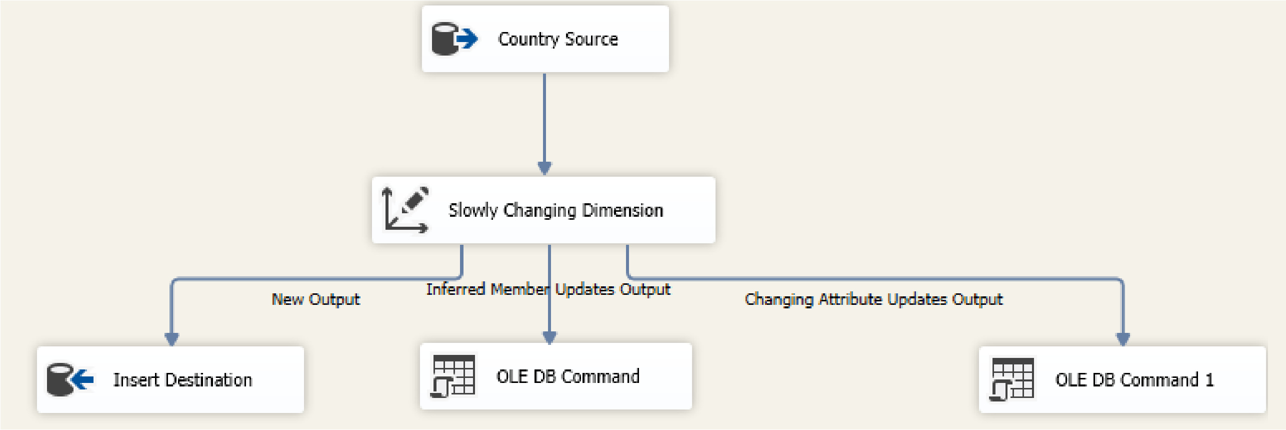
2.8 SalesPerson Table

To acquire the full data about the SalesPerson, a lookup with Employee and Person tables was performed, a derivate column to place the concatenation of first, middle and last name was used.



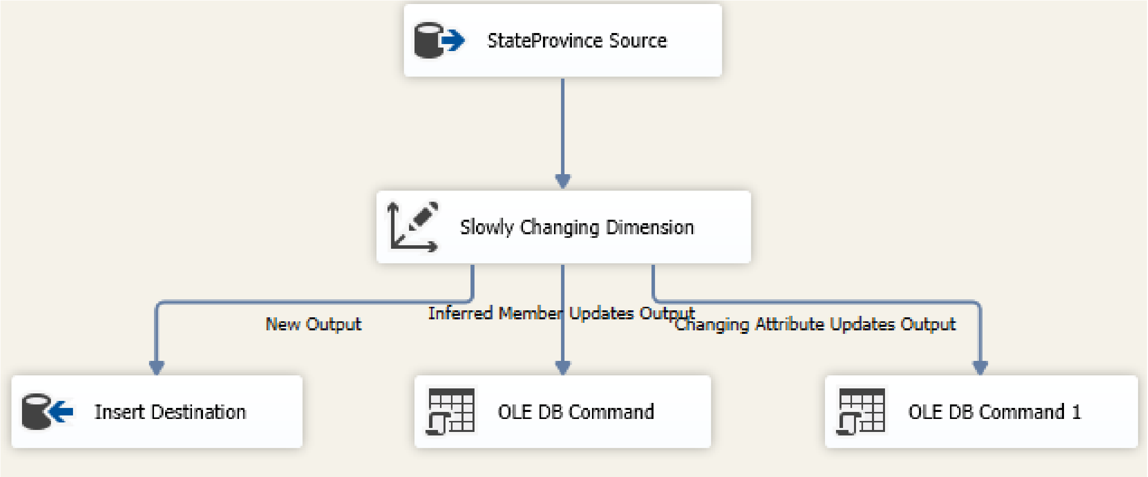
2.9 Country

Simply mapping of CountryRegionCode and Name.


Country Source
Slowly Changing Dimension
New Output
Inferred Member Updates Output
Changing Attribute Updates Output
Insert Destination
OLE DB Command
OLE DB Command 1

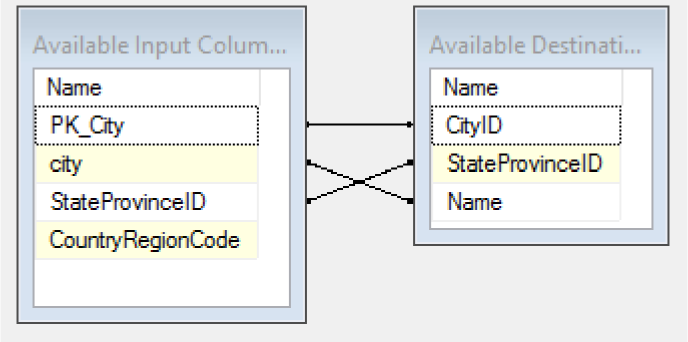
2.10 StateProvince

Mapping of Province_Name, ProvinceID and CountryRegionID


StateProvince Source
Slowly Changing Dimension
New Output
Inferred Member Updates Output
Changing Attribute Updates Output
Insert Destination
OLE DB Command
OLE DB Command 1

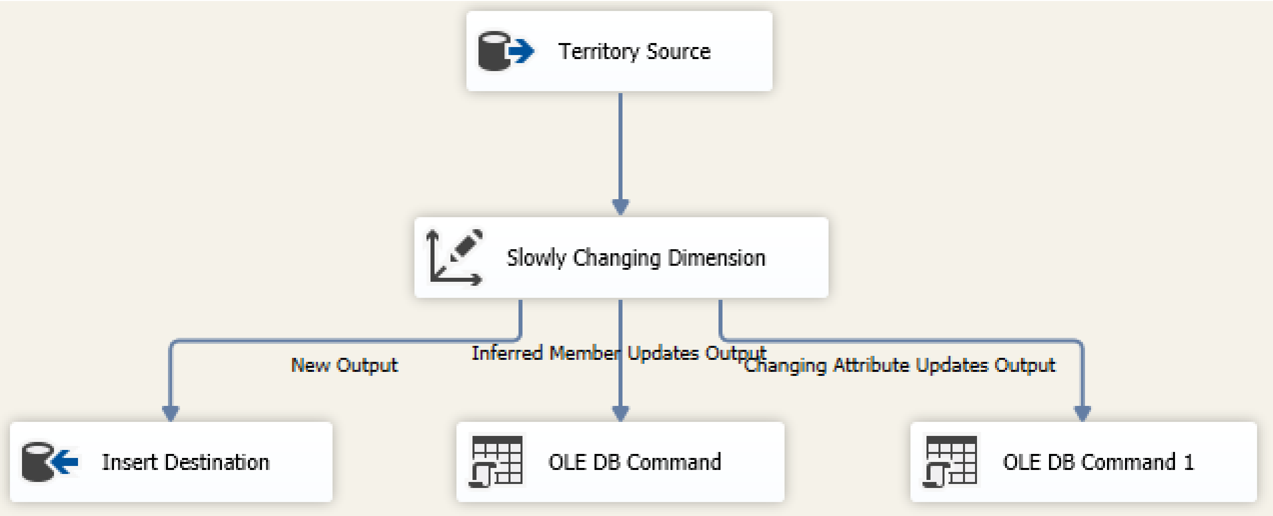
2.11 City

Simple mapping of CityID, Name and StateProvinceID



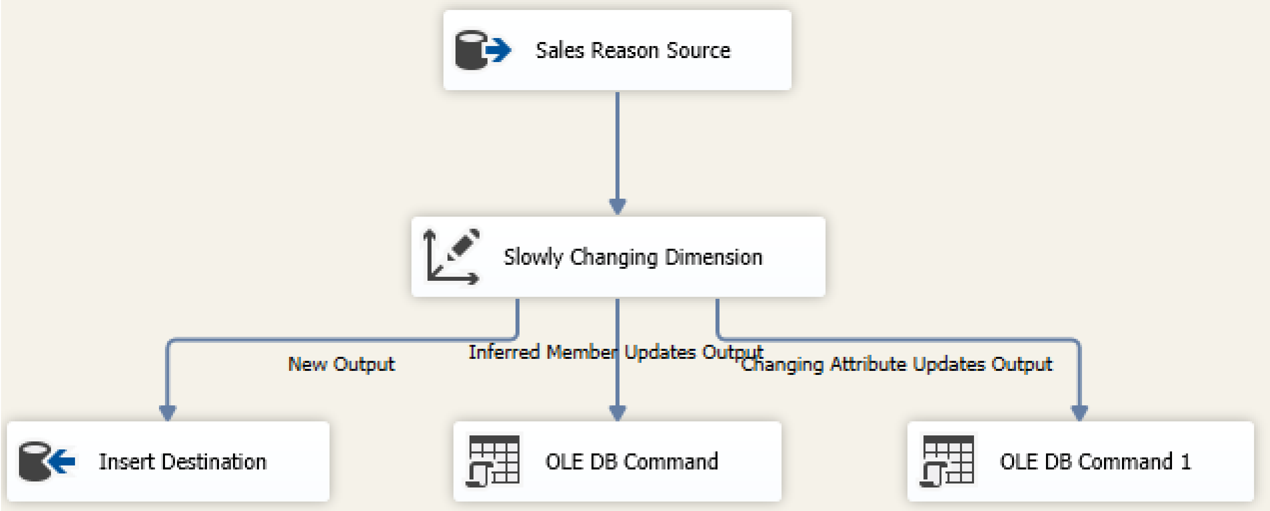
2.12 Territory

No operations were performed, aside from the direct mapping of TerritoryID, Territory Name, Group and CountryRegion code.


Territory Source
Slowly Changing Dimension
New Output
Inferred Member Updates Output
Changing Attribute Updates Output
Insert Destination
OLE DB Command
OLE DB Command 1

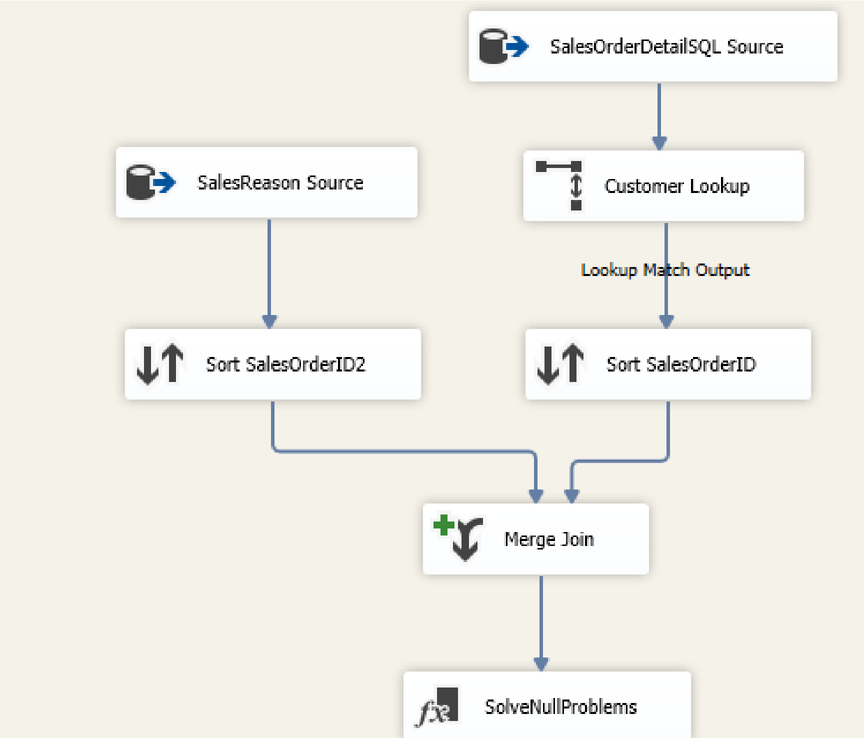
2.13 SalesReason table

A simple mapping was performed as well:


Sales Reason Source
Slowly Changing Dimension
New Output
Inferred Member Updates Output
Changing Attribute Updates Output
Insert Destination
OLE DB Command
OLE DB Command 1

2.14 The FactTable

Two sources were used: SalesReason and a query of SalesOrderDetail, the query for the second source was necessary to provide the actualized values of StandardCost and ListPrice at the sale moment. After collecting the data from SalesReason, the entries were sorted by SalesOrderID, the same thing was done for the rows collected from the query to SalesOrderDetail but before a lookup with the Customer table was done. The two sources were then merged using "Merge Join" and a derivative column to replace eventual null values of PersonID, StoreID, SalesReasonID and FK_SalesPerson was used. This measure revealed necessary because for instance some sales orders were made through internet or no reason for sales were registered.


SalesOrderDetailSQL Source
SalesReason Source
Customer Lookup
Lookup Match Output
Sort SalesOrderID2
Sort SalesOrderID
Merge Join
SolveNullProblems

3.IMPLEMENTATION OF SLOW CHANGE DIMENSIONS

3.1 Type chosen

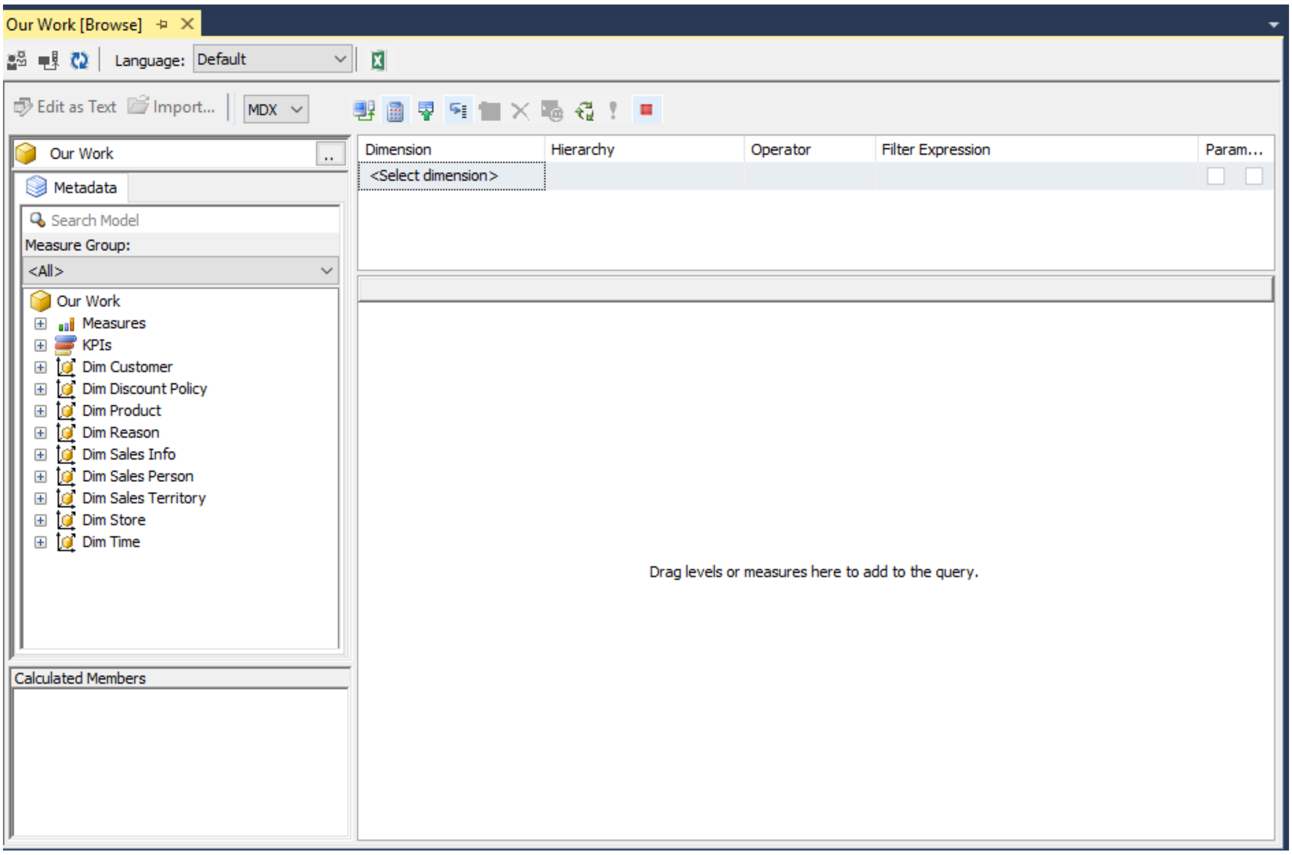
Type 1 was chosen, the new data are simply overwriting the older ones, this can be done because the data warehouse is an OLAP and does not have to obey to all the OLTP relational database rules.

3.2 Implementation of the slow changing dimension

Visual Studio Slow Changing dimension tool was used, and for each table in which is implemented the key and the attributes were identified through the wizard.

4.CONCLUSIONS

The ETL process was carried out successfully, all rows belonging to the AdventureWorksw2016CTP3 database were included in the designed database, called our_work. Since the slowly changing dimension is enabled the datawarehouse can update the designed tables when further data are inserted.
The generated Datacube process successfully and is browsable in the Analisys Service of SSMS (SQL Server Management Studio).


Our Work [Browse]
Language: Default
Edit as Text
Import...
MDX
Our Work
Metadata
Search Model
Measure Group:
<All>
Our Work
Measures
KPIs
Dim Customer
Dim Discount Policy
Dim Product
Dim Reason
Dim Sales Info
Dim Sales Person
Dim Sales Territory
Dim Store
Dim Time
Calculated Members
Dimension
Hierarchy
Operator
Filter Expression
Param...
<Select dimension>
Drag levels or measures here to add to the query.