# Lab 2: Implementing a Data Warehouse integrated in a full BI flow

## I. Setting

You are going to implement an entire BI flow according to the architecture presented in Figure 1. To do so, you will need to follow a certain sequence of steps numbered from 1 to 5. For this lab, we will use data from **a single source** which is the transactional database AdventureWorks2014. It is the DB of a fictitious bicycle manufacturer that includes data about Manufacturing, Sales, Purchasing, Product Management, Contact Management, and Human Resources.

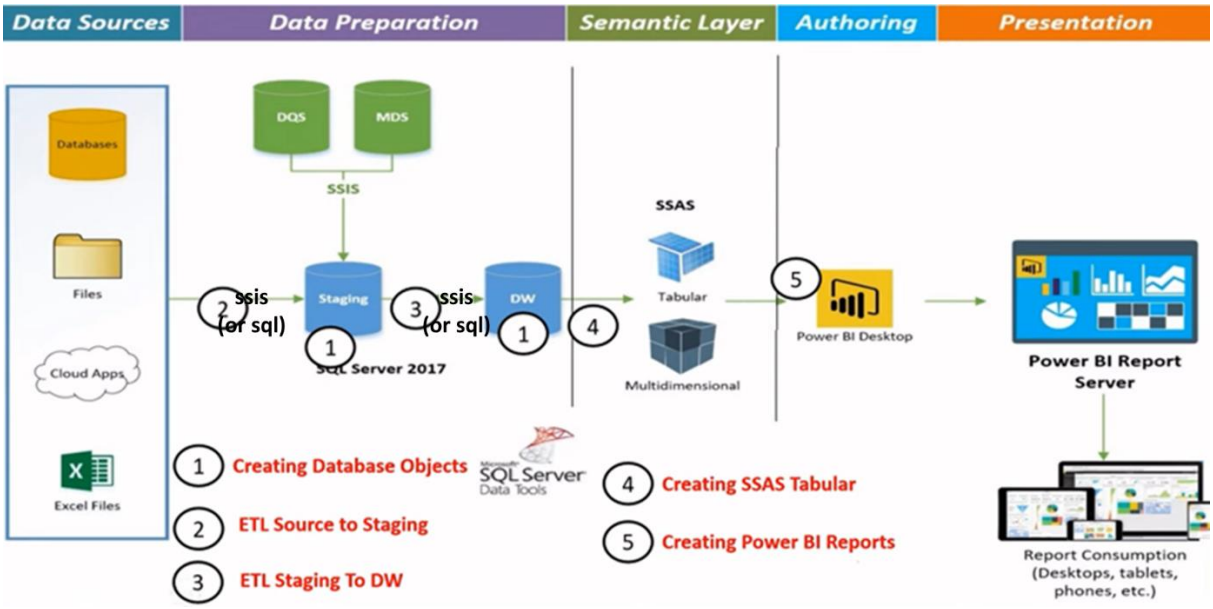


***Figure 1.*** *Architecture of the Data Warehouse and BI flow to be implemented.*

The DW to build will content data loaded from AdventureWorks2014, and will be composed of two Data Marts: Sales and HR (optional). The Data Mart Sales will be modelled according to a Galaxy Schema with two fact tables: Internet Sales and Reseller Sales. Figure 2 represents the schema of the DW you are going to build.

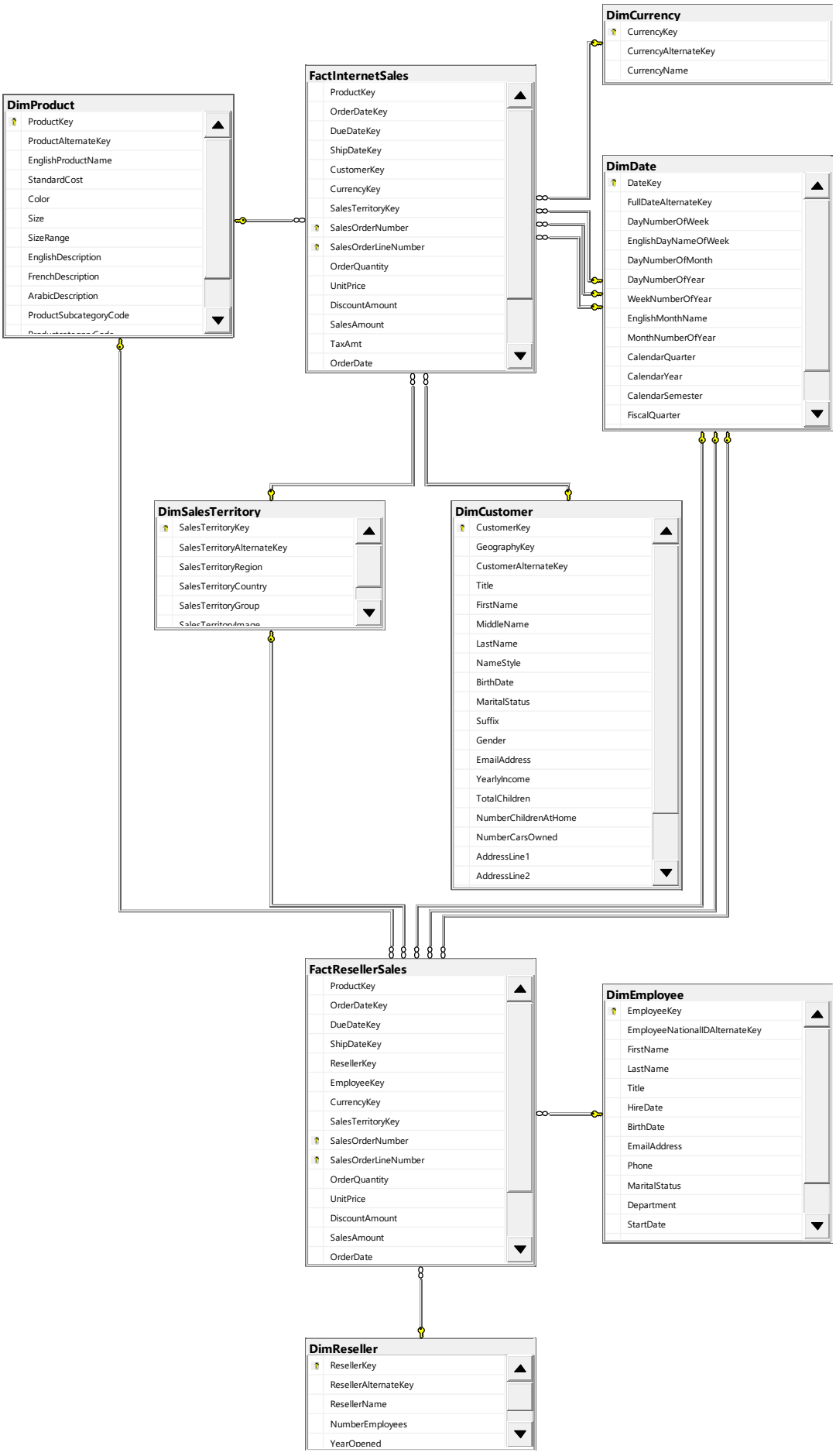
DimCurrency
CurrencyKey
CurrencyAlternateKey
CurrencyName
FactInternetSales
ProductKey
OrderDateKey
DueDateKey
ShipDateKey
CustomerKey
CurrencyKey
SalesTerritoryKey
SalesOrderNumber
SalesOrderLineNumber
OrderQuantity
UnitPrice
DiscountAmount
SalesAmount
TaxAmt
OrderDate
DimProduct
ProductKey
ProductAlternateKey
EnglishProductName
StandardCost
Color
Size
SizeRange
EnglishDescription
FrenchDescription
ArabicDescription
ProductSubcategoryCode
DimDate
DateKey
FullDateAlternateKey
DayNumberOfWeek
EnglishDayNameOfWeek
DayNumberOfMonth
DayNumberOfYear
WeekNumberOfYear
EnglishMonthName
MonthNumberOfYear
CalendarQuarter
CalendarYear
CalendarSemester
FiscalQuarter
DimSalesTerritory
SalesTerritoryKey
SalesTerritoryAlternateKey
SalesTerritoryRegion
SalesTerritoryCountry
SalesTerritoryGroup
DimCustomer
CustomerKey
GeographyKey
CustomerAlternateKey
Title
FirstName
MiddleName
LastName
NameStyle
BirthDate
MaritalStatus
Suffix
Gender
EmailAddress
YearlyIncome
TotalChildren
NumberChildrenAtHome
NumberCarsOwned
AddressLine1
AddressLine2
FactResellerSales
ProductKey
OrderDateKey
DueDateKey
ShipDateKey
ResellerKey
EmployeeKey
CurrencyKey
SalesTerritoryKey
SalesOrderNumber
SalesOrderLineNumber
OrderQuantity
UnitPrice
DiscountAmount
SalesAmount
OrderDate
DimEmployee
EmployeeKey
EmployeeNationalIDAlternateKey
FirstName
LastName
Title
HireDate
BirthDate
EmailAddress
Phone
MaritalStatus
Department
StartDate
DimReseller
ResellerKey
ResellerAlternateKey
ResellerName
NumberEmployees
YearOpened

***Figure 2.*** *Galaxy-based schema of the AdventureWorks DW. Note that the diagram is composed of two fact tables (Internet Sales and Reseller Sales). These two facts share the same dimension tables (DimSalesTerritory, DimDate, DimProduct, DimCurrency) but three of them (DimCustomer, DimReseller, DimEmployee). DimCustomer is connected to FactInternetSales, only. DimReseller and DimEmployee are linked to FactResellerSales, only.*

## II. Implementation

### *II.1. Creating Database Objects*

Step 1 consists in creating two databases. The first one is for 'Staging' (traditional relational database) and the second one is the DW per see (database modeled according to a Star/Galaxy Schema). To do so, you will have to implement the tasks as follows:

a) The staging database is a copy of the source database AdventureWorks2014. So, your first task will be to Restore AdventureWorks2014 in SSMS.

b) Understand and execute the SQL scripts '1_AWN_STG_Demo', which will create the appropriate schema, tables and views (e.g., transforming tables from the data source) for the staging DB.

c) Understand and execute the SQL script '2_AWN_DW_Demo' (until Ln 326), which will create the schema and tables of the Data Warehouse. **DO NOT EXECUTE THE CODE ABOUT THE PROCEDURES, FOR NOW (this part will be used to populate the tables).**

### *II.2. ETL Source to Staging*

In step 2, you will load the data from the source to the staging. To do so, you will have to create an SSIS package in Visual Studio, which will use AdventureWorks2014 as data source and AWN_STG_Demo as destination. The package will be composed of several Data Flow Tasks you will have to implement and execute (see Figure 3, as an example of Data Flow).

For the Sale Data Mart, you will need to populate the tables as follows: Business_Entity, Currency, Customer, Person, PersonAddress, Product, ProductCategory, ProductSubCategory, SalesHeader, SalesOrderDetails, SalesTerritory, and Store. For the HR Data Mart, you will need to populate the tables: Employee, EmployeeDepartmentHistory, and EmployeePayHistory.

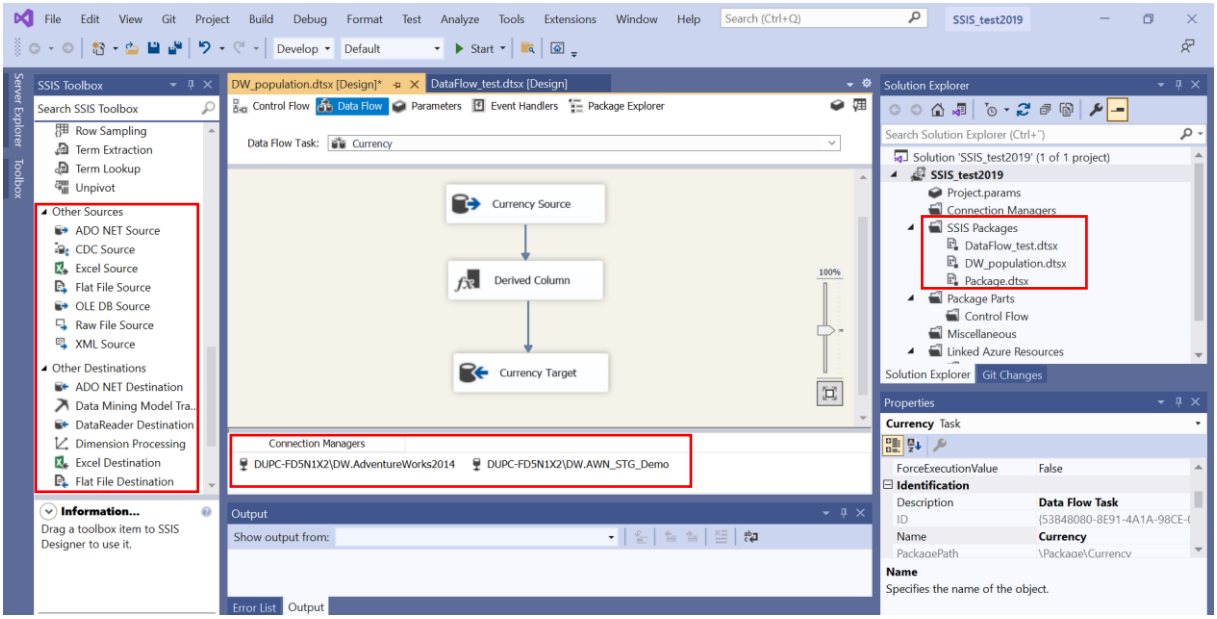


***Figure 3.*** *Example of a Data Flow implementation for the dimension Currency. Note that the loading requires a connection of the project solution to both the AdventureWorks2014 and AWN_STG_Demo databases. Choose OLE DB Source and Destination as Source and Target, respectively. The function Derived Column is used to populate columns of the target table which are not present in the source table (e.g., Created_Dt).*

Once all the data flow tasks have been executed in Visual Studio, check in SSMS if the tables of AWN_STG_Demo have been correctly populated.

*II.3. ETL Staging to DW*

In this stage, data are loaded from the staging DB to the DW. This operation has to follow the chronological order which consists in populating first the dimension tables then, second, the fact tables.

a) Start by populating the dimension tables, because the primary key must be created first. To do so, you will have to understand and **execute all the Procedures** implemented in the script 2_AWN_DW_Demo.
Note: In the script 1_AWN_STG_Demo, a View has been created for the dimension Product, in order to merge the three source tables 'Product', 'ProductCategory', and 'ProductSubCategory' (in AWN_STG_Demo) in a single table called DimProduct (in AWN_DW_Demo).
**Important:** Once you have created the procedures, do not forget to call/execute each of them to populate the dimension tables. It can be done manually in SSMS (Figure 4) or through an Execute SQL Task with SSIS.

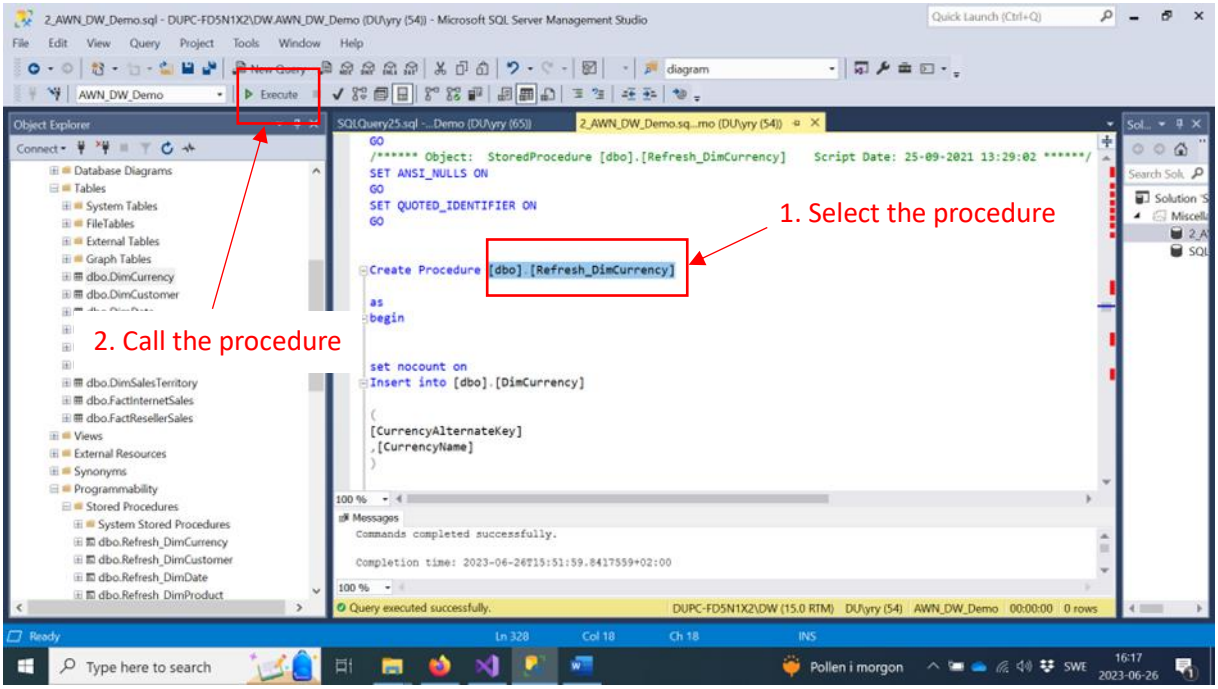


***Figure 4.*** *Manual execution of a procedure, once the it has been created through an SQL script (example for the Currency dimension).*

Now, all the dimension tables should be populated, but the DimEmployee dimension.

b) The **DimEmployee** has to be populated differently, because it needs to **maintain history**, as it is illustrated in Figure 5, where the Supplier ABC moved from CA to IL after 2014-11-22. In other words, we need to keep a track of changes of certain attributes (here is 'State') that may occur over the time.

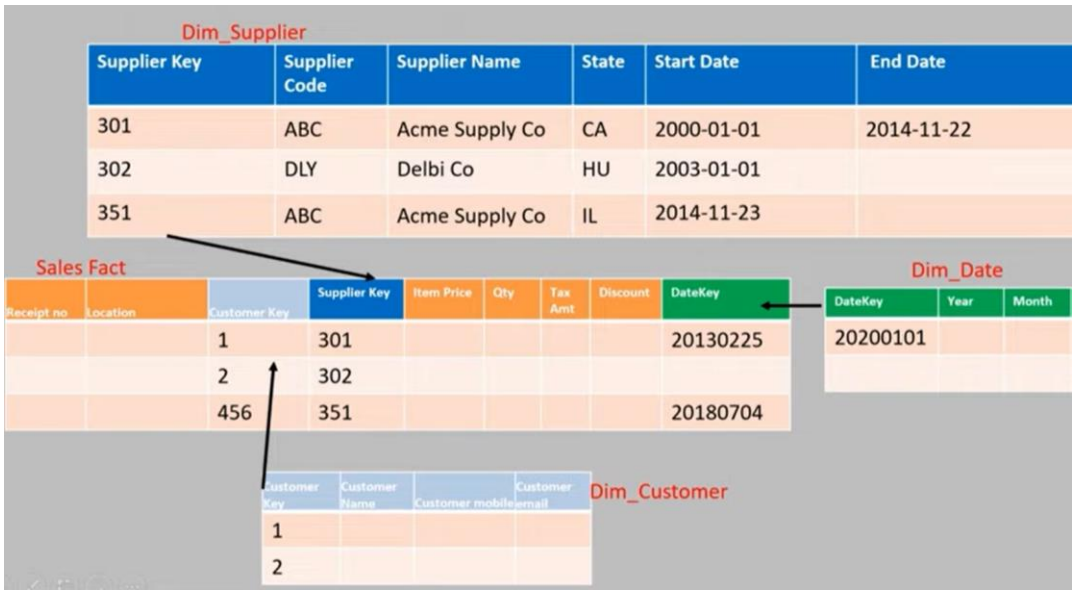


***Figure 5.*** Example of three dimensions *(Dim_Supplier, Dim_Date, and Dim_Customer)* with their respective surrogate key (Supplier Key, Date Key, and Customer Key), which are referenced in the fact table *(Sales Fact)*. Besides the foreign keys, the fact table contents measures.

To maintain history, you need to create and execute a new SSIS package which implements a Data Flow Task including a 'Slowly Changing Dimension' as illustrated in Figure 6. The OLE DB Source must be Stg_vw_Erp_Employee. The configuration of the Slowly Changing Dimension should be as follows:

- Target table is DimEmployee.
- Indicate the right label for the missing 'Input Columns', if any.
- NationalIDNumber must be Business key as 'Key Type'.
- MaritalStatus must be configurated as 'Changing attribute', whereas Title and Department must be configurated as 'Historical attribute'.
- Select 'StartDate' and 'EndDate', and create these variables (System:CreationDate).

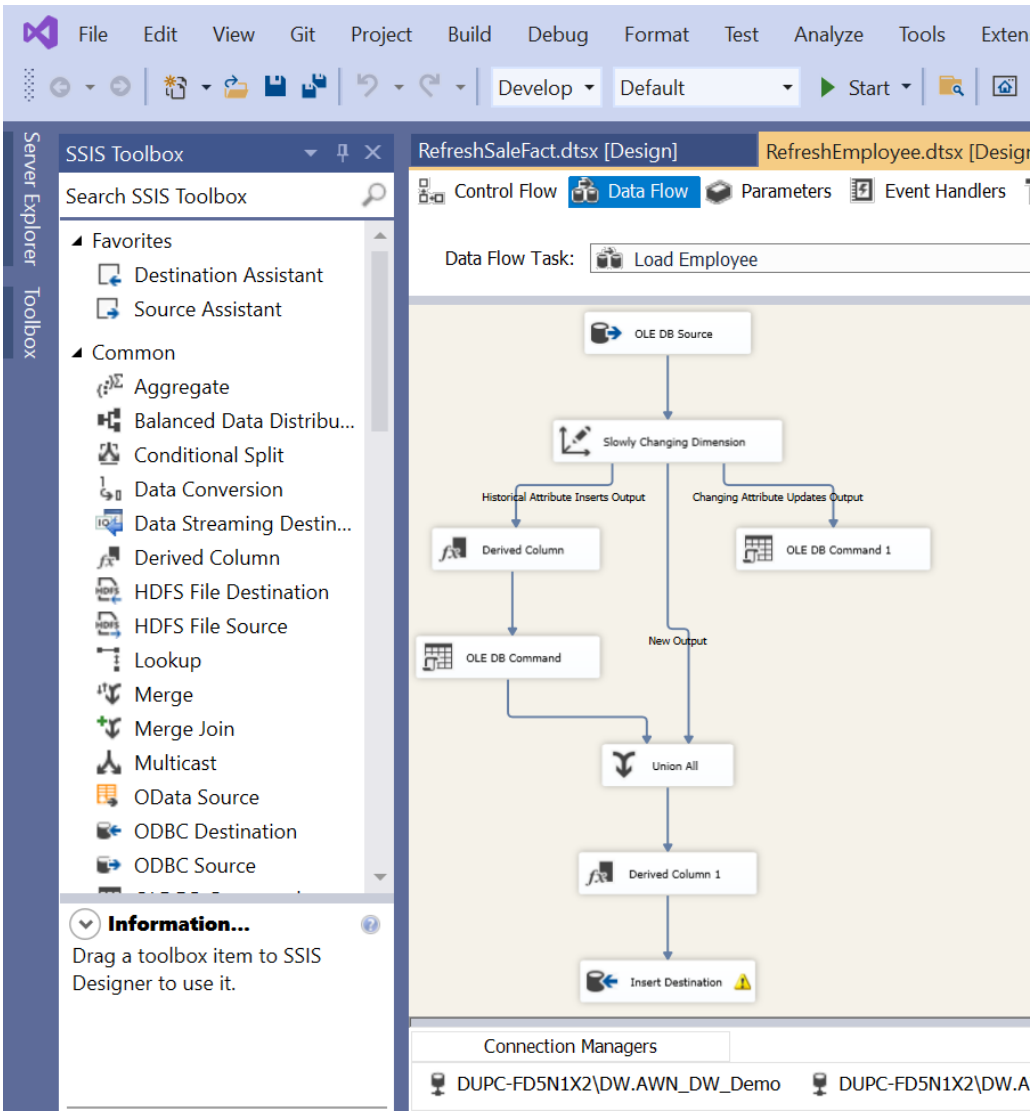


***Figure 6.*** *Data Flow Task which handles both historical (left branch of the diagram) and changing (right branch of the diagram) attributes. Note that once you have configurated the Slowly Changing Dimension, the remaining flow will be created automatically.*

c) Finally, you will have to populate the two fact tables. To do so, you will have to create two Data Flow Tasks, one for each fact table. Start with the Internet Sales Fact, then you will repeat the same process for Reseller Sales Fact. Here are the steps you need to follow:
- Create an OLE DB Source which will read the View called Stg_vw_Erp_Fact_InternetSales.
- Create 'Lookup' transformations to join the fact table to the dimension tables through a foreign key. Figure 7 shows an example for ProductKey.

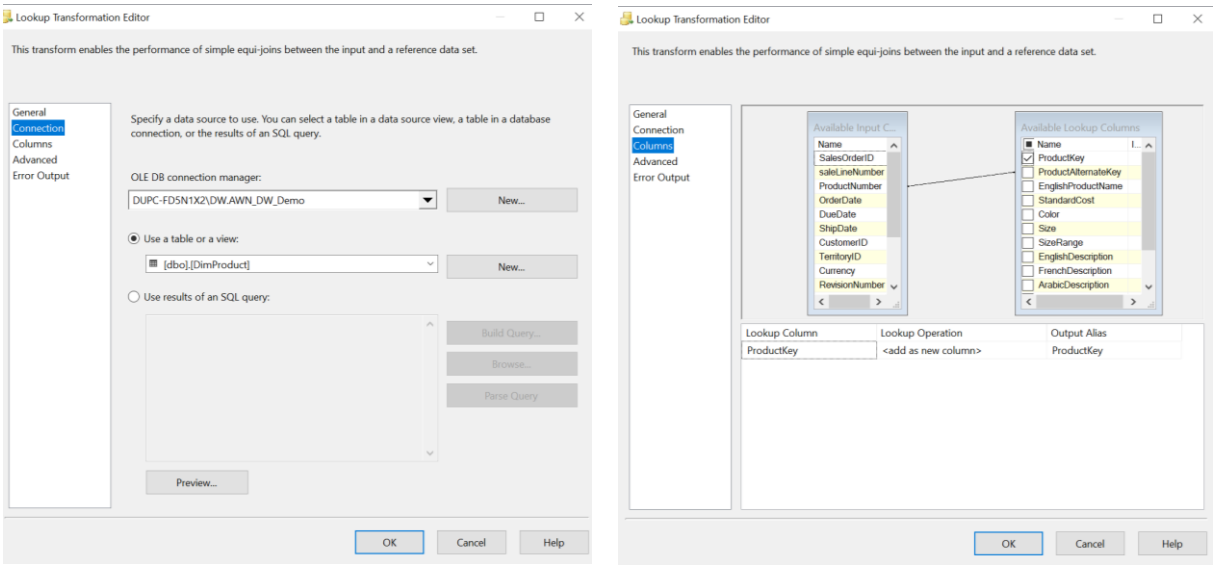


***Figure 7.*** *Configuration of the Lookup transformation to map the ProductAlyernativeKey, from the dimension table, with the ProductNumber, from the fact table.*

- Repeat the same process for the other dimensions: OrderDateKey, DueDateKey, ShipDateKey, CurrencyKey, CustomerKey, and TerritoryKey.
- Finally, you create an OLE DB Destination to load the data in the FactInternetSales table (Figure 8). Note that if there is no matching with the Input Column, you need to map it manually (Figure 8, left panel).

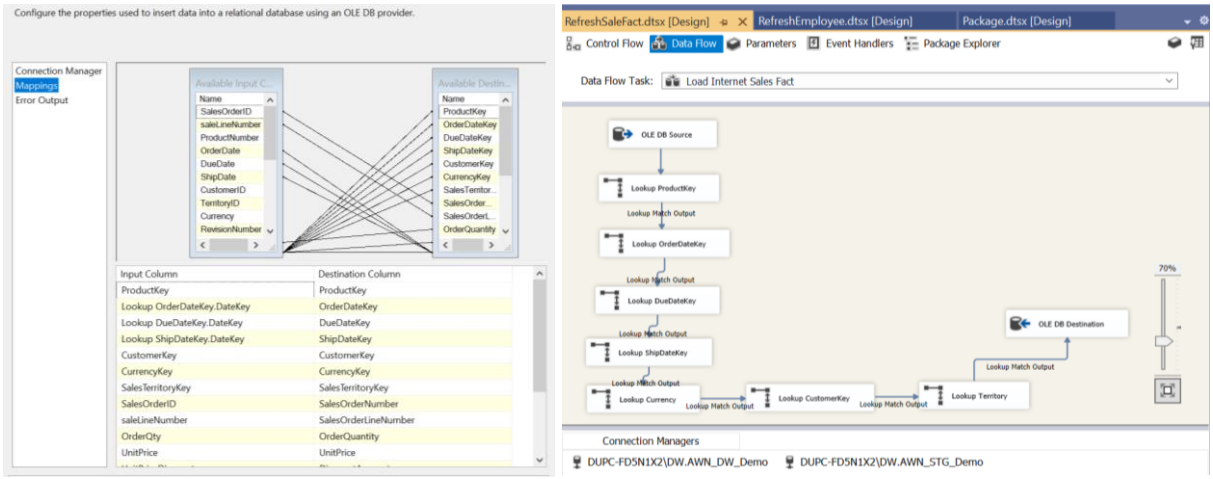


***Figure 8.*** *Mapping between the Internet Fact table and its respective dimensions (left panel); and overview of the Data Flow Task to get such a mapping (right panel).*

- Once everything is configurated correctly in the SSIS package, you can execute the task to populate the two fact tables. Then, check in SSMS that all the data are now loaded in the DW.

*II.4. Creating SSAS tabular*

Step 4 consists in implementing a tabular cube. To do so, you need to proceed as follows:
- First, make sure that the extension 'Analysis Service' is installed in your version of Visual Studio.
- Create a new Analysis Service Tabular Project. During the configuration select the appropriate Workspace **Analysis Server** to deploy your project.

- In the Tabular Model Explorer of your project, import the data from your DW. CAREFUL when you configurate the **Service Account**!!! To make it possible, you have to add the 'SQL Server Analysis' (DW, in Figure 9) account as 'New Login' in SSMS.

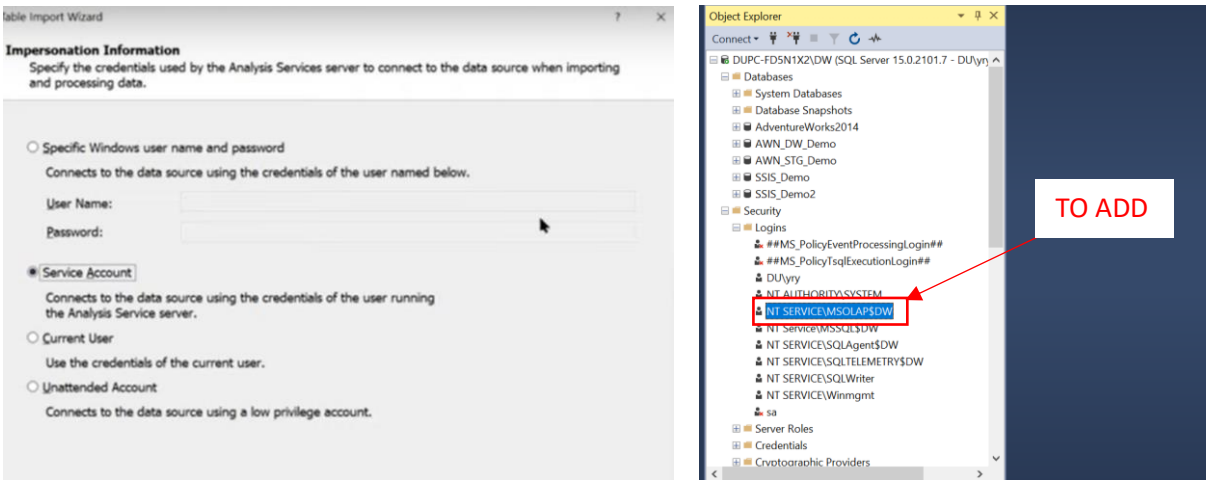


***Figure 9.*** *Check that the SQL Server Analysis account is added as Login, in order to have the permission to read and import data from the DW.*

- Select the two fact tables and related dimension tables of the Sales Data Mart. Check that the tables and galaxy schema as been correctly imported in SSAS project.
- Once everything is done, you can (i) build and deploy the solution and (ii) see the result in SSMS by connecting to the Analysis Services Server (Figure 10).

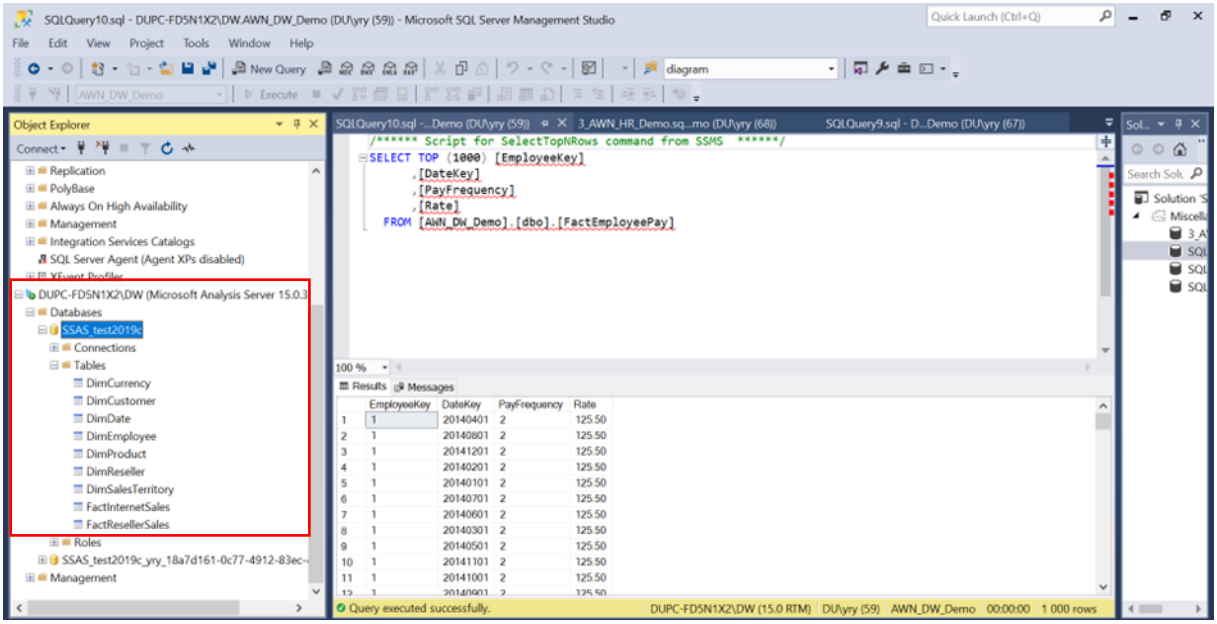


***Figure 10.*** *It is important you notice that the cube is not located in the same server as the DW. AWN_DW_Demo is hosted by the Database Engine Server, whereas the cube is hosted by the Analysis Services Server.*

*II.5. Creating Power BI Reports*

Finally, you will have to issue a Power BI report based on the data gotten from the SSAS Cube (Figure 11).

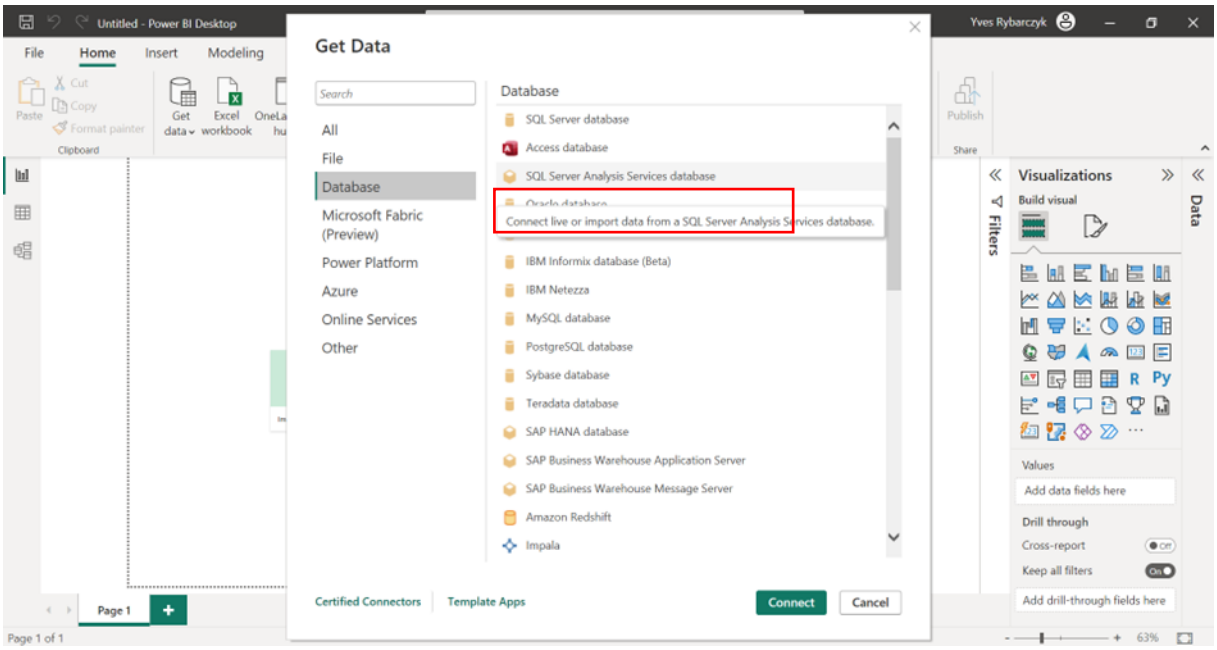


***Figure 11.*** *Get Data from the SQL Server Analysis Services which hosts the SSAS cube.*

The report should provide the features as follows:
- All possible analyses of the Sales according to the following combinations: 3 years (2005 | 2006 | 2007) x 12 months (Jan – Dec) x 4 Product categories (Bikes | Components | Clothing | Accessories) x Product subcategories (Mountain Bikes, Road Bikes, Touring Bikes, …) x 6 Countries (US | CA | FR | DE | AU | GB) x 3 Continents (North America | Europe | Pacific) x 2 Modalities (Internet | Reseller).
- Cumulative statistics: Total Sales, Average Monthly Sales, Average Quarterly Sales, Average Weekly Sales, and Average Daily Sales.

Feel free to use your proper design for the report (interactive charts, dynamic output, …) as long as you display the information requested above.

**III. Software applications to install**

- SQL Server Engine: https://www.microsoft.com/en-us/sql-server/sql-server-downloads

- SQL Server Management Studio (SSMS): https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16

- Visual Studio data tools, including SSIS and SSAS (Community version): https://visualstudio.microsoft.com/downloads/

- Power BI Desktop: https://www.microsoft.com/en-us/download/details.aspx?id=58494

- AdventureWorks2014.bak (OLTP version): https://learn.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver16&tabs=ssms

- Three SQL Scripts for creating the Staging DB, the DW, and the HR Data Mart: https://drive.google.com/file/d/1-oEOH-CHCmuyVKCWCPXceLsqUXmO0_Xo/view

**IV. Advanced tasks if you want to go further…**

- Create the Data Mart for HR.

- Deploy your SSIS project solutions and create an SQL Server Agent to automatize the DW update according to a specific schedule.

- Produce a Power BI report for a multidimensional analysis of the HR data.

**V. Deliverables**

- Short report summarizing your implementation of the five stages (include screenshots of the key steps and final schema of the DW).

- New scripts and/or project solutions you implemented (e.g., SQL code, SSIS packages, SSAS tabular cube, …).

- Power BI desktop report.

**VI. Schedule**

*VI.1. Session 1*

- Installation of the software applications.

- Creating Database Objects.

*VI.2. Session 2*

- ETL Source to Staging.

- ETL Staging to DW.

*VI.3. Session 3*

- Creating SSAS Tabular.

- Creating Power BI report(s).

*VI.4. Assessment*

Each group will have 20 minutes to present its work and answer my questions for evaluating the individual contribution of each team member.