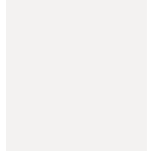
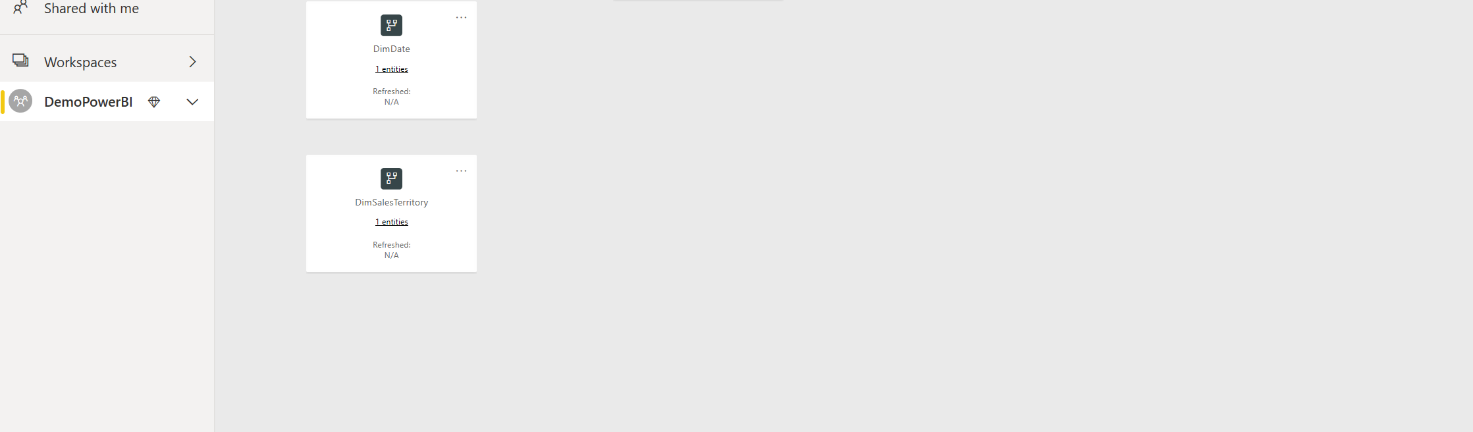
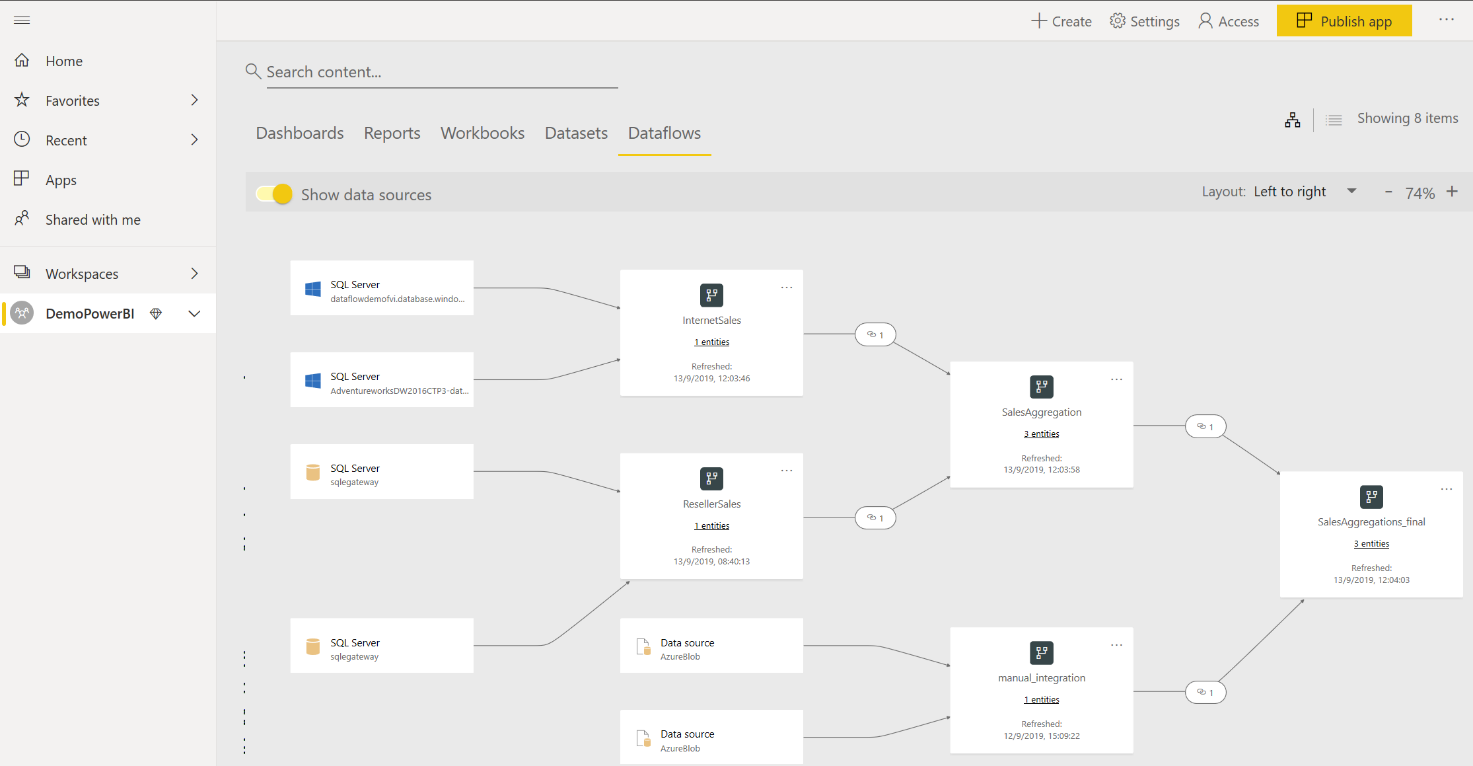
**Power BI Dataflows – Demo Script**



1. Environment configuration:
   1. **Portale Azure**: Power BI Embedded capacity creation (when the service on the portal, choose Tier A1 ) 🡪 <https://docs.microsoft.com/en-us/power-bi/developer/azure-pbie-create-capacity>

**Jump this step if Power BI Premium capacity is already allocated**

* 1. **Power BI Service**: Access section Admin Portal-> Capacity settings and select tab "Power BI Embedded". By accessing with same Azure portal credentials, the user can configure capacity created at 1. and activate the use of Power BI Dataflows service 🡪 <https://docs.microsoft.com/en-us/power-bi/service-admin-premium-manage>
  2. Download **Azure Storage Explorer** (it will be useful to set permission between Data Lake and Power BI Service at step 4.) 🡪 <https://azure.microsoft.com/it-it/features/storage-explorer>
  3. **Portale Azure + Power BI Service**: Data Lake creation and Power BI workspace configuration to allow the import Dataflow folders in Azure Data Lake Gen2 🡪 <https://docs.microsoft.com/en-us/power-bi/service-dataflows-connect-azure-data-lake-storage-gen2>
  4. **Power BI Service**: create a workspace by activating dedicated capacity e dataflow storage 🡪 <https://docs.microsoft.com/en-us/power-bi/service-dataflows-configure-workspace-storage-settings>

1. Data source setting:
   1. Main data sources: .csv available in the folder or entire database available as **AdventureworksDW2016CTP3.bacpac** imported in Azure SQL Database and in SQL Server in a VM in this example (to simulate an on prem environment).
      * FactInternetSales
      * FactResellerSales
   2. manual\_integration source: Integration.csv loaded in a Blob Storage to simulate the connection to a file outside the official systems.
   3. DimDate and DimSalesTerritory added as external CDM folder to Power BI as a dataflow (folders, available within the demo materials) have to be loaded in the data lake and then follow the documentation to retrieve the model.json path link to be inserted <https://docs.microsoft.com/en-us/power-bi/service-dataflows-add-cdm-folder>

|  |
| --- |
| let  Source = Sql.Databases("dataflowdemofvi.database.windows.net"),  Navigation = Source{[Name = "AdventureworksDW2016CTP3"]}[Data],  #"Navigation 1" = Navigation{[Schema = "dbo", Item = "FactInternetSales"]}[Data],  #"Choose columns" = Table.SelectColumns(#"Navigation 1", {"ProductKey", "OrderDateKey", "SalesTerritoryKey", "TotalProductCost", "SalesAmount", "OrderDate", "ShipDate"}),  #"Changed column type" = Table.TransformColumnTypes(#"Choose columns", {{"TotalProductCost", type number}, {"SalesAmount", type number}}),  #"Multiplied Column" = Table.TransformColumns(#"Changed column type", {{"TotalProductCost", each \_ \* -1, type number}}),  #"Inserted addition" = Table.AddColumn(#"Multiplied Column", "GrossProfit", each [SalesAmount] + [TotalProductCost], type number),  #"Unpivoted Columns" = Table.UnpivotOtherColumns(#"Inserted addition", {"ProductKey", "OrderDateKey", "SalesTerritoryKey", "OrderDate", "ShipDate"}, "Attribute", "Value"),  #"Renamed columns" = Table.RenameColumns(#"Unpivoted Columns", {{"Attribute", "Category"}, {"Value", "Amount"}}),  #"Changed column type 1" = Table.TransformColumnTypes(#"Renamed columns", {{"Amount", Currency.Type}}),  #"Inserted Year" = Table.AddColumn(#"Changed column type 1", "Year", each Date.Year([OrderDate]), Int64.Type),  #"Added Custom Column" = Table.AddColumn(#"Inserted Year", "Custom", each Text.Combine({DateTime.ToText([OrderDate], "yyyy"), DateTime.ToText([OrderDate], "MM")}), type text),  #"Renamed Columns" = Table.RenameColumns(#"Added Custom Column", {{"Custom", "YearMonth"}}),  #"Inserted Literal" = Table.AddColumn(#"Renamed Columns", "Literal", each "Internet", type text),  #"Renamed Columns1" = Table.RenameColumns(#"Inserted Literal", {{"Literal", "Channel"}})  in  #"Renamed Columns1" |

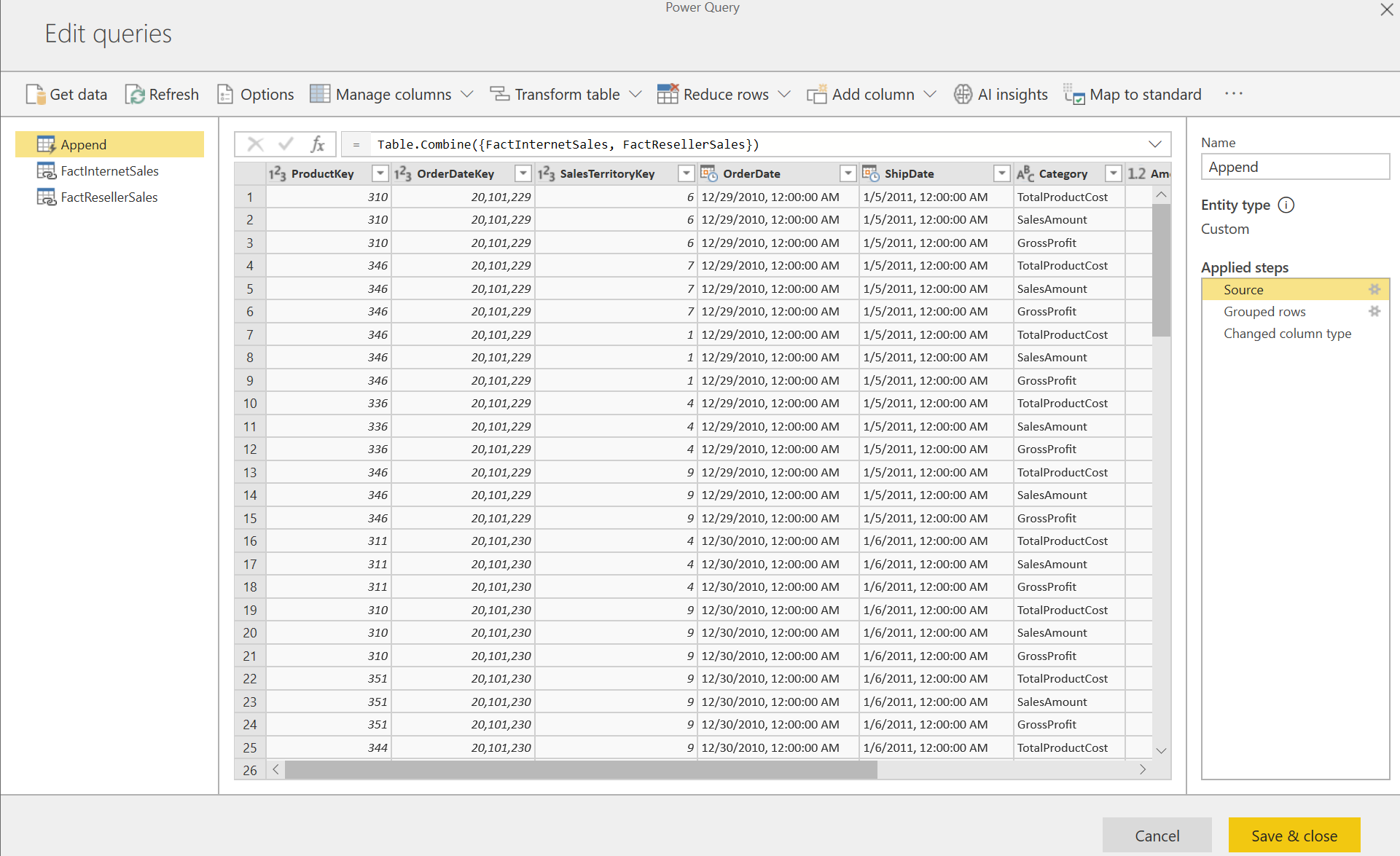
1. Create a dataflow for Internet and one for Reseller Sales and apply the following transformations (Source depends on the environment you are setting):
   1. FactInternetSales:
   2. FactResellerSales:

|  |
| --- |
| let  Source = Sql.Databases("sqlegateway"),  Navigation = Source{[Name = "AdventureworksDW2016CTP3"]}[Data],  #"Navigation 1" = Navigation{[Schema = "dbo", Item = "FactResellerSales"]}[Data],  #"Choose columns" = Table.SelectColumns(#"Navigation 1", {"ProductKey", "OrderDateKey", "SalesTerritoryKey", "TotalProductCost", "SalesAmount", "OrderDate", "ShipDate"}),  #"Changed column type" = Table.TransformColumnTypes(#"Choose columns", {{"TotalProductCost", type number}, {"SalesAmount", type number}}),  #"Multiplied Column" = Table.TransformColumns(#"Changed column type", {{"TotalProductCost", each \_ \* -1, type number}}),  #"Inserted addition" = Table.AddColumn(#"Multiplied Column", "GrossProfit", each [SalesAmount] + [TotalProductCost], type number),  #"Unpivoted Columns" = Table.UnpivotOtherColumns(#"Inserted addition", {"ProductKey", "OrderDateKey", "SalesTerritoryKey", "OrderDate", "ShipDate"}, "Attribute", "Value"),  #"Renamed columns" = Table.RenameColumns(#"Unpivoted Columns", {{"Attribute", "Category"}, {"Value", "Amount"}}),  #"Changed column type 1" = Table.TransformColumnTypes(#"Renamed columns", {{"Amount", Currency.Type}}),  #"Inserted Year" = Table.AddColumn(#"Changed column type 1", "Year", each Date.Year([OrderDate]), Int64.Type),  #"Added Custom Column" = Table.AddColumn(#"Inserted Year", "Custom", each Text.Combine({DateTime.ToText([OrderDate], "yyyy"), DateTime.ToText([OrderDate], "MM")}), type text),  #"Renamed Columns" = Table.RenameColumns(#"Added Custom Column", {{"Custom", "YearMonth"}}),  #"Inserted Literal" = Table.AddColumn(#"Renamed Columns", "Literal", each "Reseller", type text),  #"Renamed Columns1" = Table.RenameColumns(#"Inserted Literal", {{"Literal", "Channel"}})  in  #"Renamed Columns1" |

* 1. SalesAggregations created with the concept of Linked entity between FactInternetSales and FactResellerSales <https://docs.microsoft.com/en-us/power-bi/service-dataflows-linked-entities>

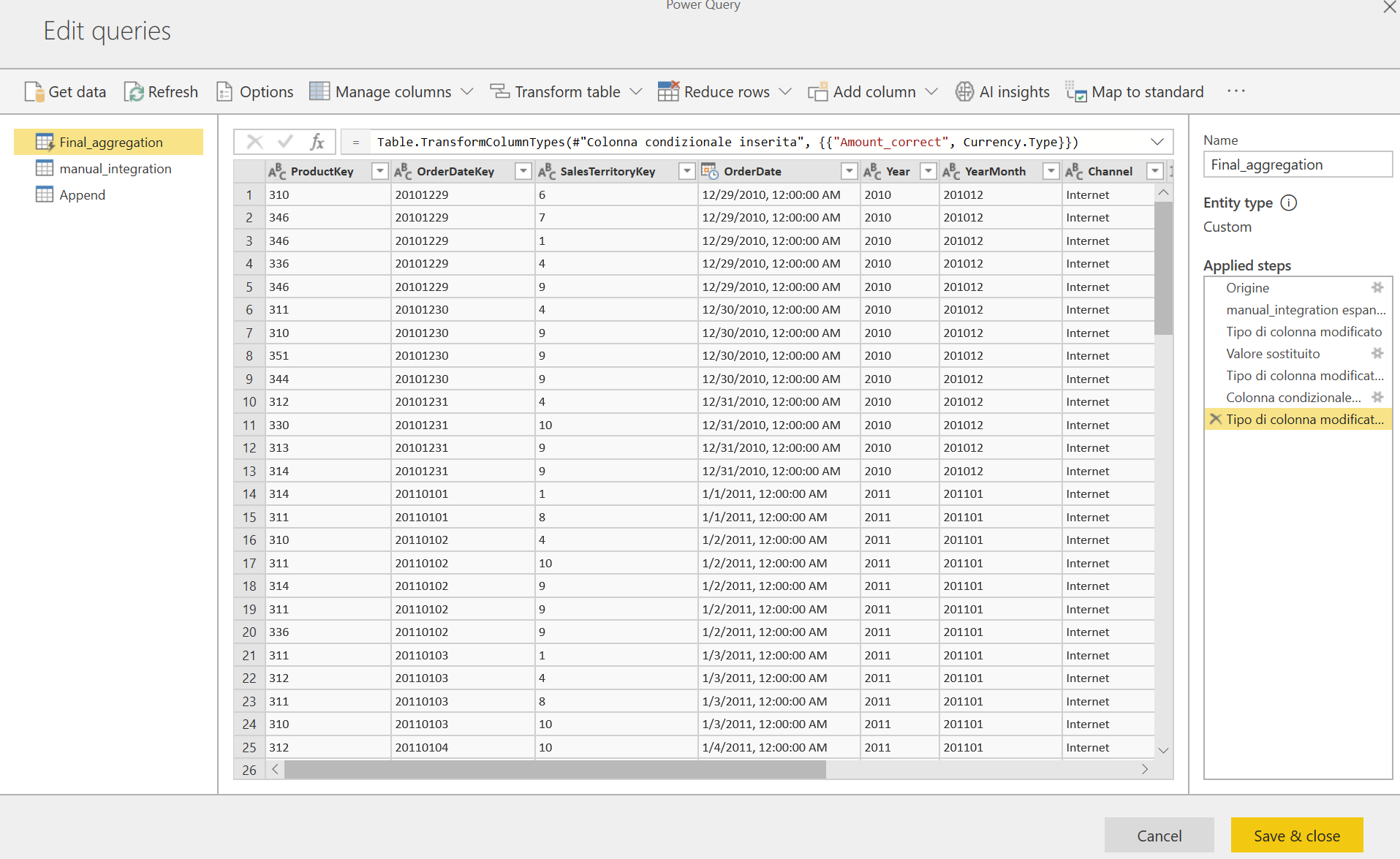
Once the Linked entity is created, an “Appen Query as new” transformation was applied:

1. Create a Dataflow 🡪 <https://docs.microsoft.com/en-us/power-bi/service-dataflows-create-use>

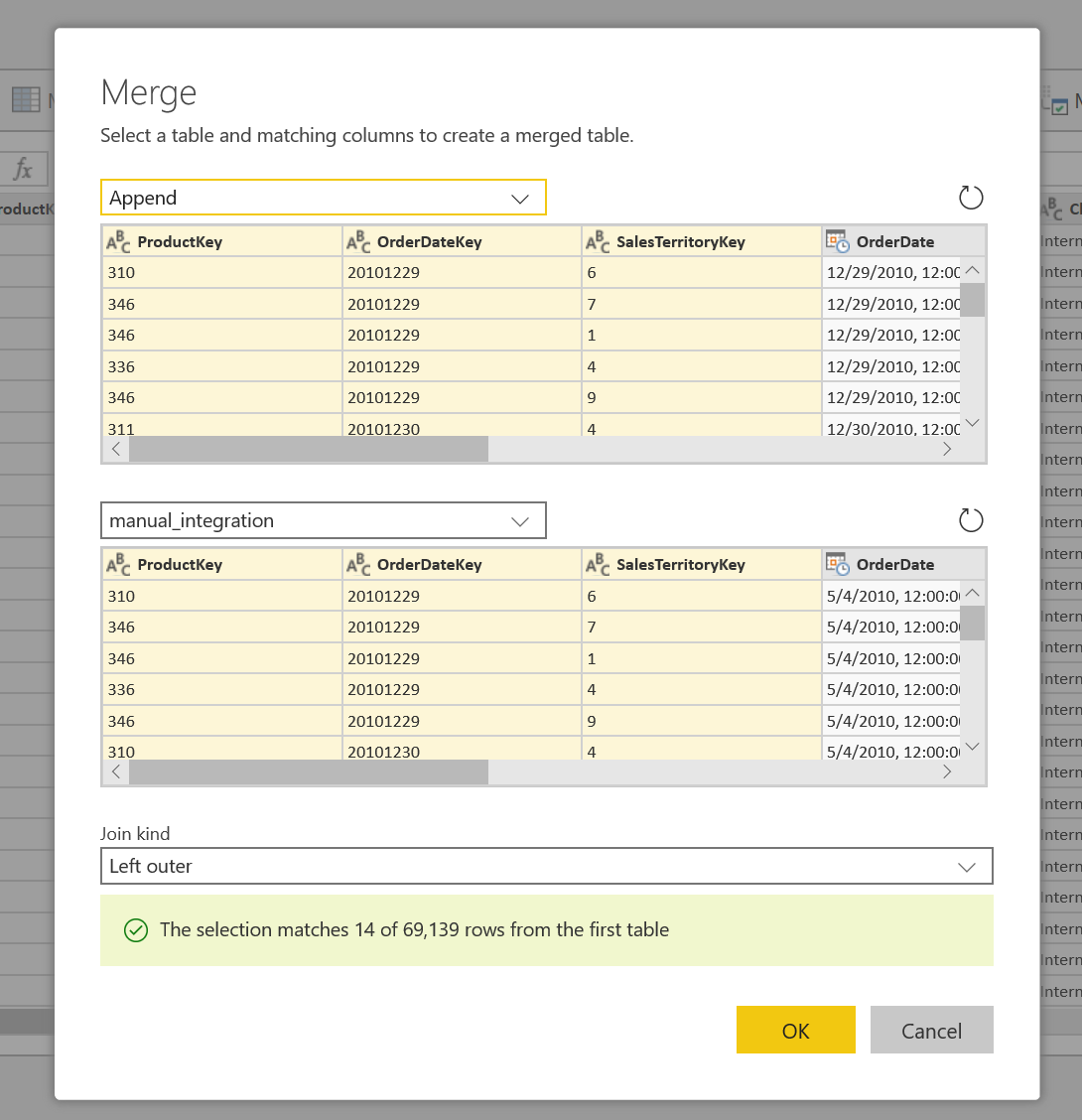


Accessing the “Advanced Editor” of Append table, the following should be the code output

|  |
| --- |
| let  Source = Table.Combine({FactInternetSales, FactResellerSales}),  #"Grouped rows" = Table.Group(Source, {"ProductKey", "OrderDateKey", "SalesTerritoryKey", "OrderDate", "Year", "YearMonth", "Channel"}, {{"Amount", each List.Sum([Amount]), type number}}),  #"Changed column type" = Table.TransformColumnTypes(#"Grouped rows", {{"OrderDate", type date}, {"ProductKey", type text}, {"OrderDateKey", type text}, {"SalesTerritoryKey", type text}, {"Year", type text}})  in  #"Changed column type" |

1. SalesAggregation\_final created with the concept of linked entity between SalesAggregations and manual\_integration. After the link is created, a “merge queries as new” transformation was applied.

Merge transformation is applied from Append table as a left outer join on Product Key, OrderDateKey, SalesTerritoryKey and Year:



When the Merge is applied, SalesAggregations\_final table is produced and some transformations are applied as in the M code below:

|  |
| --- |
| let  Origine = Table.NestedJoin(Append, {"ProductKey", "OrderDateKey", "SalesTerritoryKey", "Year"}, manual\_integration, {"ProductKey", "OrderDateKey", "SalesTerritoryKey", "Year"}, "manual\_integration", JoinKind.LeftOuter),  #"manual\_integration expanded" = Table.ExpandTableColumn(Origine, "manual\_integration", {"Amount"}, {"manual\_integration.Amount"}),  #"Changed column type" = Table.TransformColumnTypes(#"manual\_integration expanded", {{"manual\_integration.Amount", type text}}),  ReplaceValue = Table.ReplaceValue(#"Changed column type", null, "0", Replacer.ReplaceValue, {"manual\_integration.Amount"}),  #"Changed column type 1" = Table.TransformColumnTypes(ReplaceValue, {{"manual\_integration.Amount", type number}}),  #"Add Conditional column" = Table.AddColumn(#"Changed column type 1", "Amount\_correct", each if [manual\_integration.Amount] = 0 then [Amount] else [manual\_integration.Amount]),  #"Changed column type 2" = Table.TransformColumnTypes(#"Add Conditional column", {{"Amount\_correct", Currency.Type}})  in  #"Changed column type 2" |

The idea is to allow end users to overwrite data that come from systems (ex. FactInternetSales and FactResellerSales by integrating manual files that can have a different origin). After the merge, an if condition is applied by creating a Conditional Column that allows the user to integrate the data coming from the manual\_integration data source.

4. Open Power BI Desktop and connect to PBI Dataflows and import SalesaAggregation\_final and the DimTables. Create the model and start creating the report.