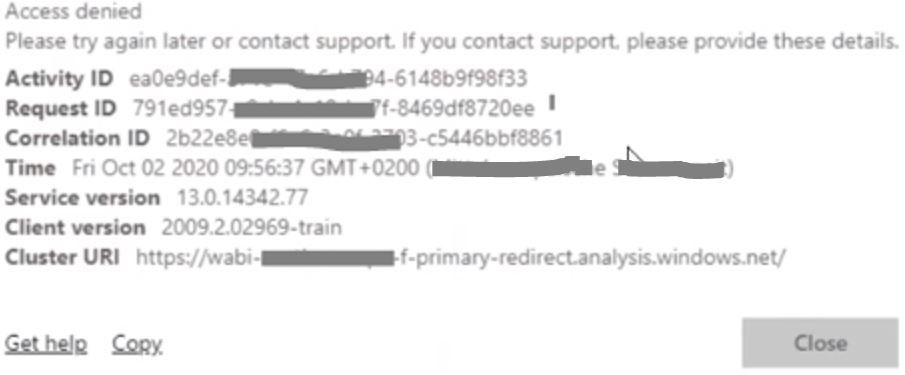
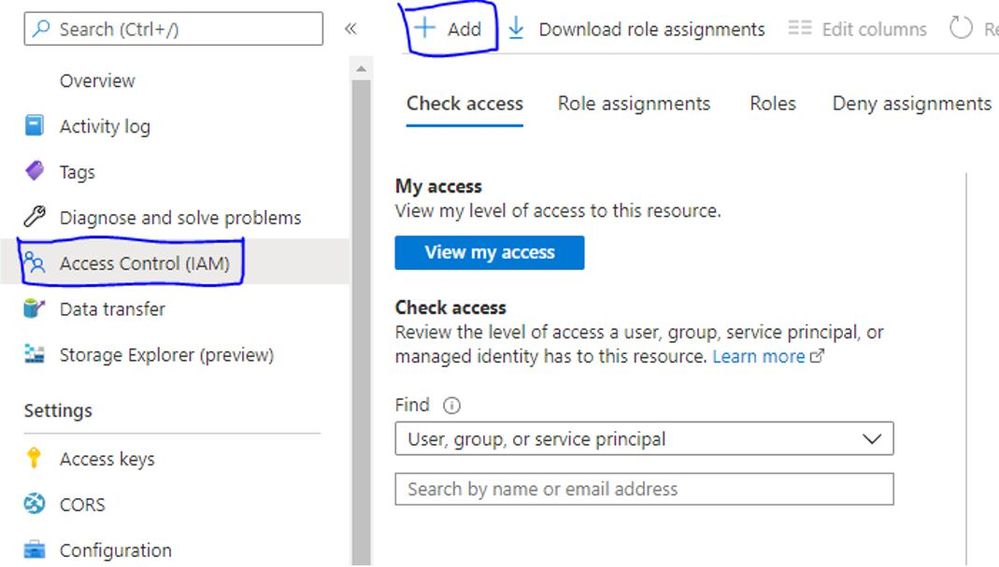
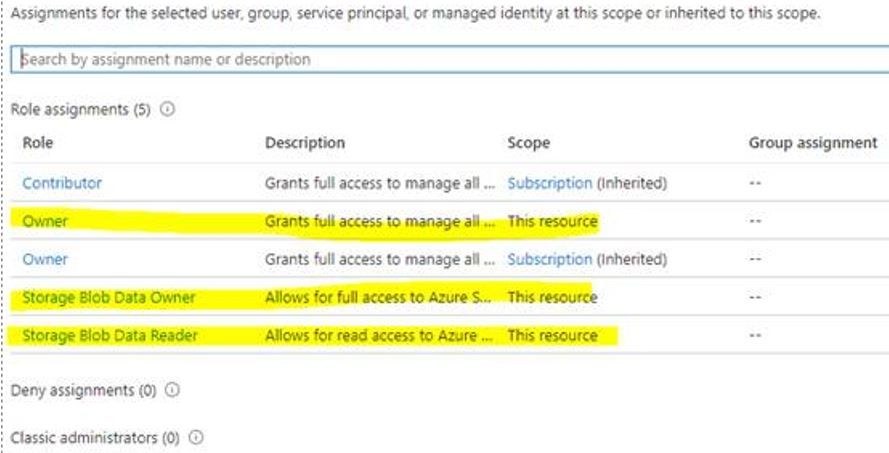
DEMO 1

1. Creare un workspace **Dataflow**.
2. Mostrare che si può assegnare un **Dataflow** ad una versione **Azure Data Lake** Personale e non usare quella di default
   1. Molto probabilmente darà errore quello che si vede sotto

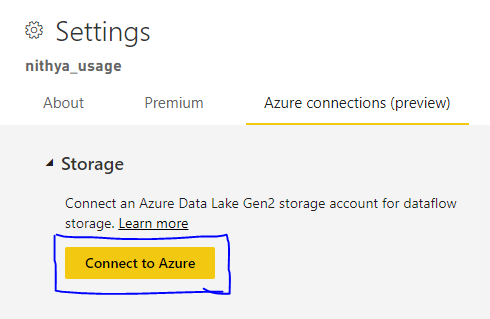


* 1. Se da errore bisogna abilitare i ruoli





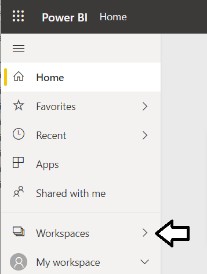
* 1. Poi si può procedere a collegare il workspace al data lake



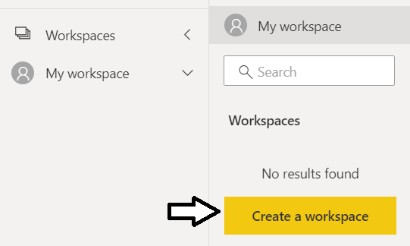
Demo 2

# Exercise 1: Use Power Query to Create a New Dataflow

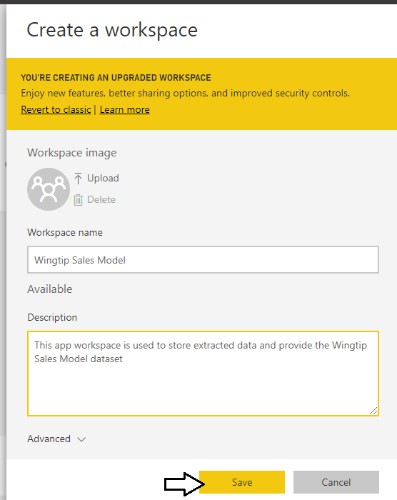
1. Log into the Power BI Service with your new organizational account.
   1. Navigate the Power BI portal at [https://app.powerbi.com](https://app.powerbi.com/) and if prompted, log in using your organizational account.
2. Create a new app workspace named **Wingtip Sales Model**.
   1. Click the **Workspace** flyout menu in the left navigation.



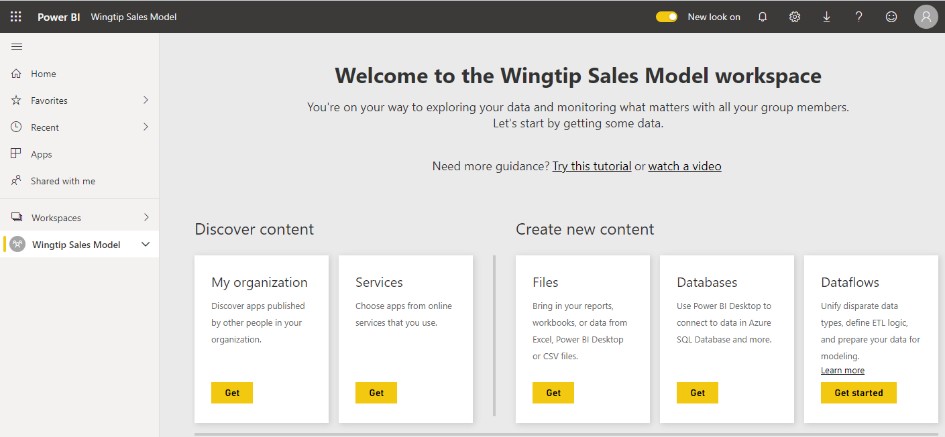
* 1. Click the **Create app workspace** button to display the **Create an app workspace** dialog.



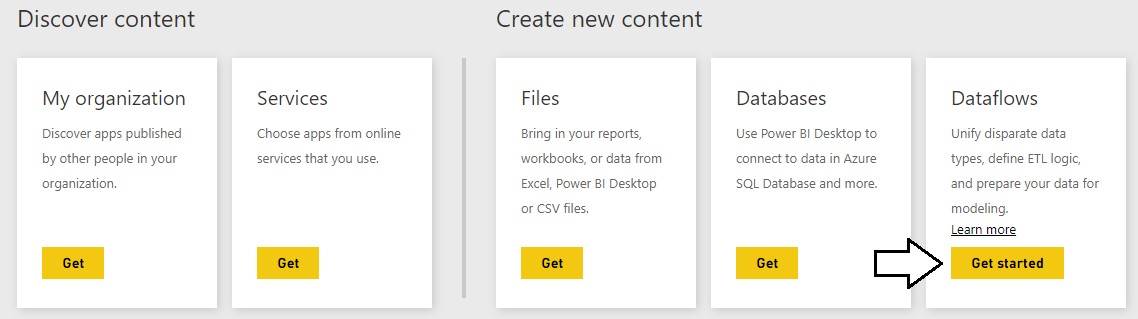
* 1. In the **Create an app workspace** pane, enter a workspace name of **Wingtip Sales Model**.
  2. Click the **Save** button to create the new app workspace named **Wingtip Sales Model**.



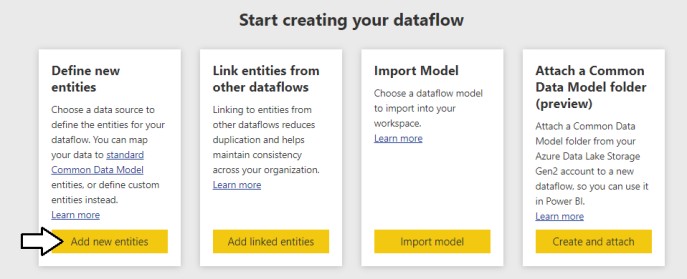
* 1. When you click **Save**, the Power BI service should create the new app workspace and then switch your current Power BI session to be running within the context of the new **Wingtip Sales Model** workspace.



1. Create a new dataflow.
   1. Click the **Get started** button in the **Dataflows** section.



* 1. Click the **Add new entities** button in the **Define new entities** section.

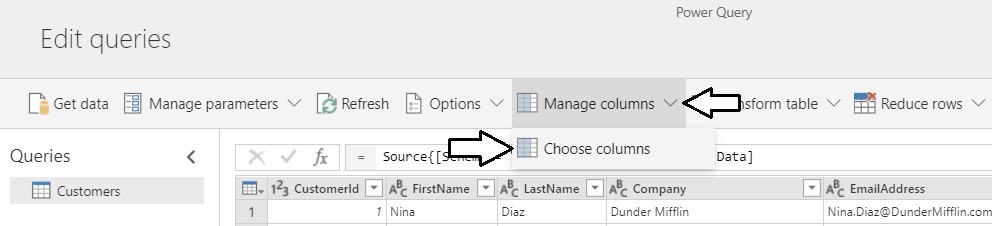


* 1. On the **Choose data source** page, select the **Database** tab and then select **SQL Server database**.

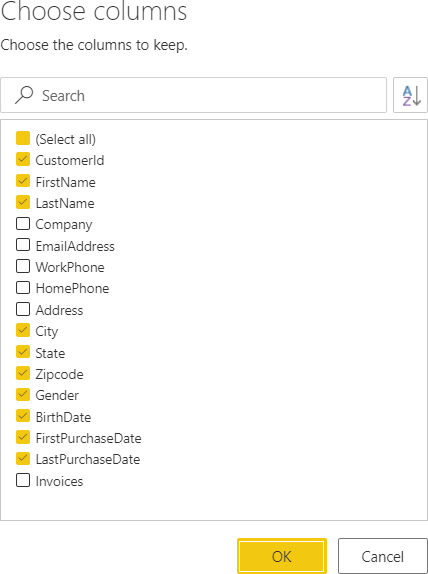
1. Select the **Customers** table https://raw.githubusercontent.com/marcopozzan/Dati/master/PBI\_Course\_Customers.csv
   1. In the **Choose data** section, select the **Customers** table and then click **Transform data**.
   2. You should now see the **Edit queries** screen displaying query results and the M formula bar for the current step.



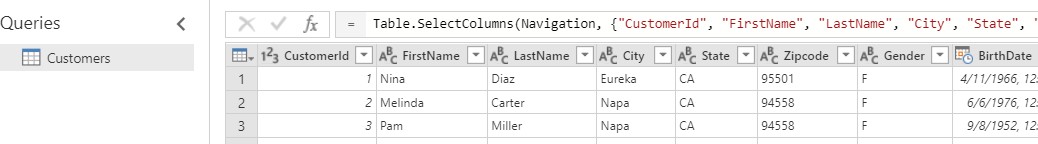
1. Use Power Query to clean and transform the data from the **Customers** table.
   1. Drop down the **Manage columns** menu button and select the **Choose columns** command.



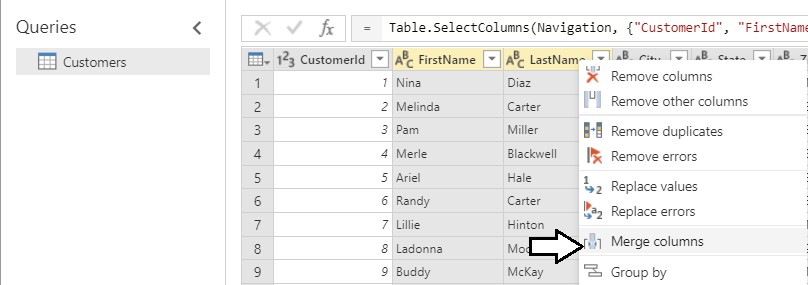
* 1. In the **Choose columns** dialog, begin by clicking on the **(Select all)** checkbox at the top to unselect all column. Next, select the checkboxes for the following columns.
     1. **CustomerId**
     2. **FirstName**
     3. **LastName**
     4. **City**
     5. **State**
     6. **Zipcode**
     7. **Gender**
     8. **BirthDate**
     9. **FirstPurchaseDate**
     10. **LastPurchaseDate**
  2. Once you have the columns selected as shown in the following screenshot, click **OK** to close the **Choose columns** dialog.



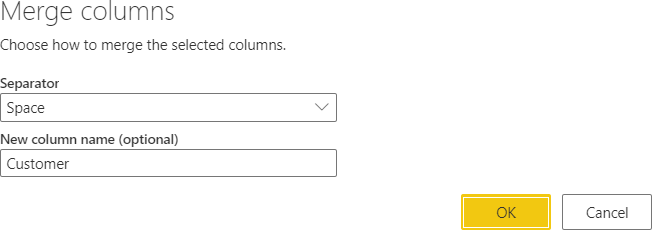
* 1. You should be able to verify that the Power Query editor now only shows the columns that you selected.



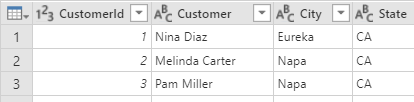
1. In this step you will merge the **FirstName** column and the **LastName** column together into a single column named **Customer**.
   1. Select the **FirstName** column by clicking on its column header.
   2. Next, hold down the **SHIFT** key and select the **LastName** column by clicking on its column header.
   3. Right-click on the selected columns and click the **Merge columns** menu command.



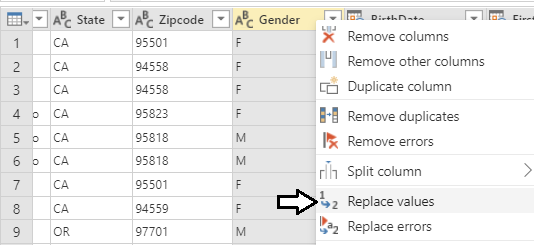
* 1. In the **Merge Column** dialog, drop down the **Separator** control and select a value of **Space**.
  2. Add a **New column name** value of **Customer** and click the **OK** button to modify the underlying query with your changes.



* 1. You should now be able to see that the **FirstName** column and the **LastName** column have been replaced with a single merged column named **Customer**.

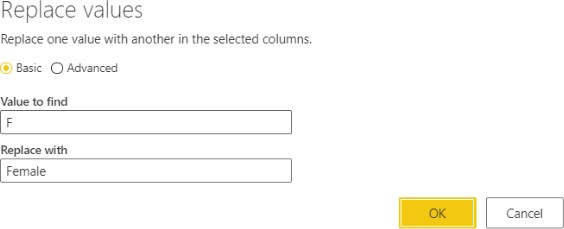


1. Modify the query so that the **Gender** column returns values of **Male** and **Female** instead of **M** and **F**.
   1. Locate the **Gender** column in the **Customers** table.
   2. Right-click the header for the **Gender** column and select the **Replace values** command to display the **Replace values** dialog.

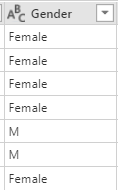


* 1. In the **Replace values** dialog, enter a value of **F** in the **Value to find** textbox and enter a value of **Female** in the **Replace with**

textbox. Click to **OK** button add your changes to the underlying query.

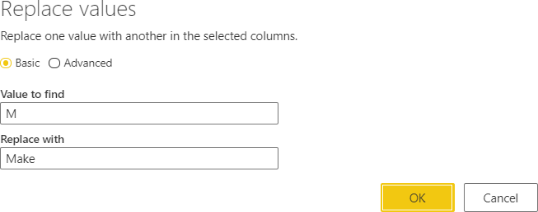


* 1. You should be able to see that all values of **F** in the **Gender** column have been replaced with a value of **Female**.



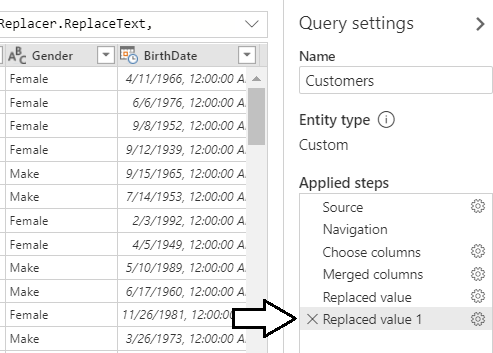
* 1. Right-click the header for the **Gender** column and select the **Replace values** command a second time.
  2. In the **Replace values** dialog, enter a value of **M** in the **Value to find** textbox and enter a value of **Male** in the **Replace with**

textbox. Click to **OK** button add your changes to the underlying query.

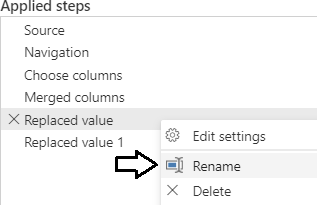


* 1. You should be able to confirm that all values in the **Gender** column have been replaced with a value of either **Male** or **Female**.

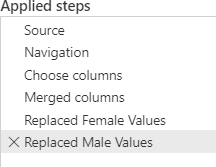
1. Change the name of query steps.
   1. Inspect the **Applied Steps** list in the **Query settings** pane. You should be able to see that there are two steps at the end that have been given the generic names of **Replaced value** and **Replaced value 1**.



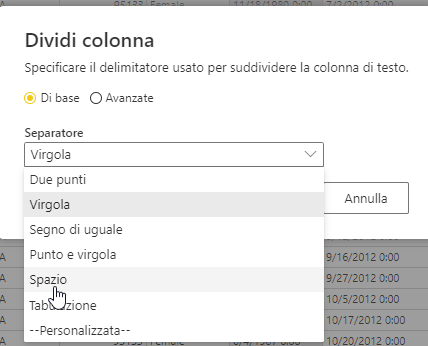
* 1. Rename the **Replaced Values** step by right-clicking it and clicking the **Rename** command to place the step name in edit mode. Modify the name of this step to **Replace Female Values**.



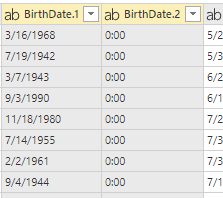
* 1. Using the same technique, rename the **Replaced Value 1** step to **Replaced Male Values**.

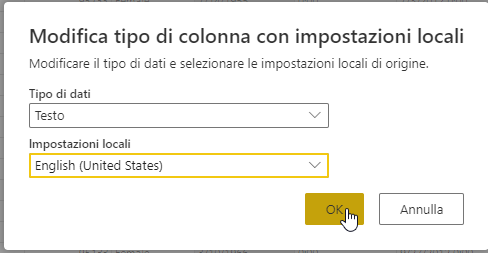


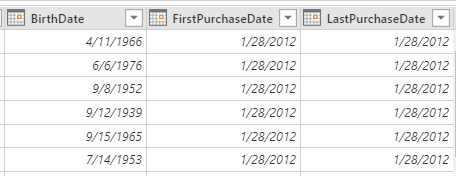
1. Change the column type of **BirthDate**, **FirstPurchasedDate** and **LastPurchasedDate** from **Date/Time** to **Date**.
   1. Dividi la colonna birthdate in base al delimitatore spazio



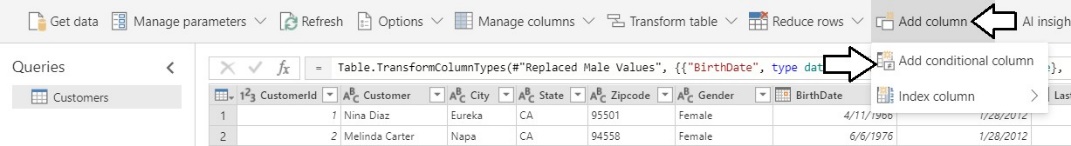
* 1. Elimina altra colonna **BirthDate.2**



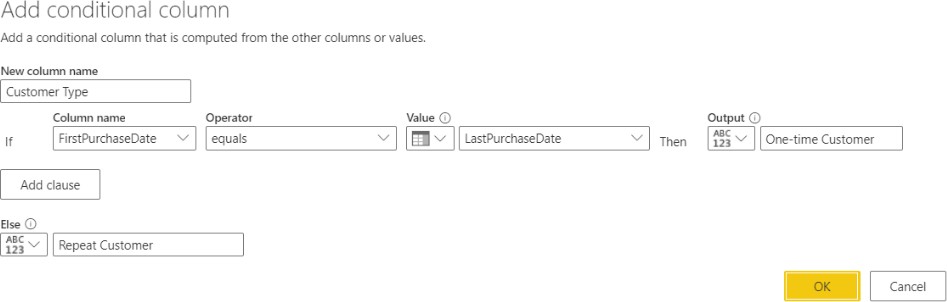
* 1. Use the column type drop down on the left-hand side of the **BirthDate** column to configure the column using the **Date** type with location
  2. Use the datatype drop down menu of the **FirstPurchaseDate** column to configure the column using the **Date** type.
  3. Use the datatype drop down menu of the **LastPurchaseDate** column to configure the column using the **Date** type.
  4. You should see that the three columns now show values with a date but without a time.



1. Add a new conditional column named **Customer Type** to indicates whether the customer is a repeat customer or not.
   1. Drop down the **Add column** menu button and select the **Add conditional column** command.



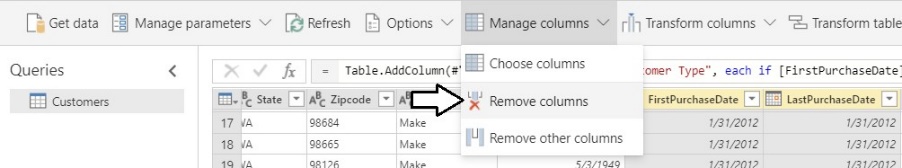
* 1. In the **Add conditional column** dialog, enter a **New column name** value of **Customer Type**.
  2. Configure a rule to return a string value of “One-time Customer” if **FirstPurchaseDate** equals **LastPurchaseDate**.
  3. For the **Else** evaluation, return a string value of “Repeat Customer”.
  4. When the **Add conditional column** dialog matches the screenshot below, click the **OK** button to add the new column.



* 1. You should be able to verify that the new **Customer Type** column has a value of **Repeat Customer** when the current customer has a **FirstPurchaseDate** column value that is not equal to the **LastPurchaseDate** column value. When these column values are equal, the **Customer Type** column has a value of **One-time Customer**.

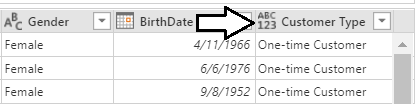


1. Remove the **FirstPurchaseDate** column and the **LastPurchaseDate** column.
   1. Select the **FirstPurchaseDate** column by clicking its column header.
   2. Hold down the **SHIFT** key and click the column header for **LastPurchaseDate** so that both columns are selected.
   3. Right click the one of the selected columns and click the **Remove Columns**.

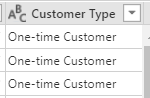


* 1. You should be able to confirm that the **FirstPurchaseDate** column and the **LastPurchaseDate** columns have been removed from the query results. However, the **Customer Type** column is still there.

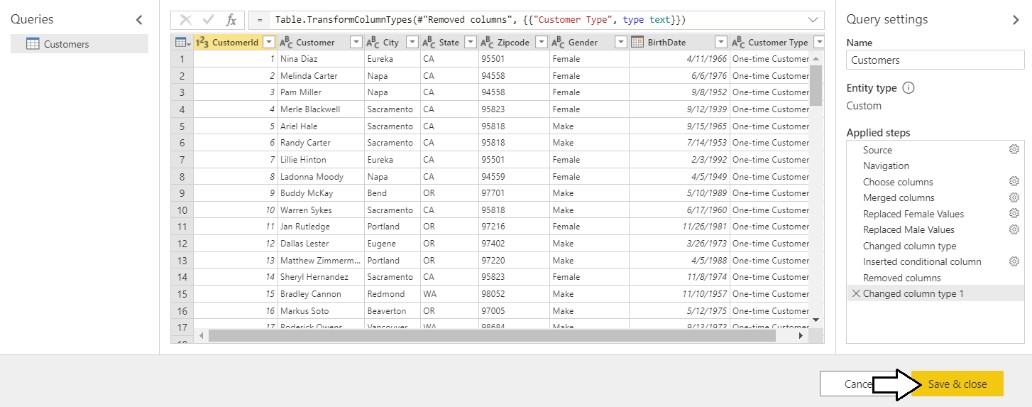
1. Set the column type for the **Customer Type** column to **Text**.
   1. You might notice that column type menu for the **Customer Type** column is not set to a specific type.



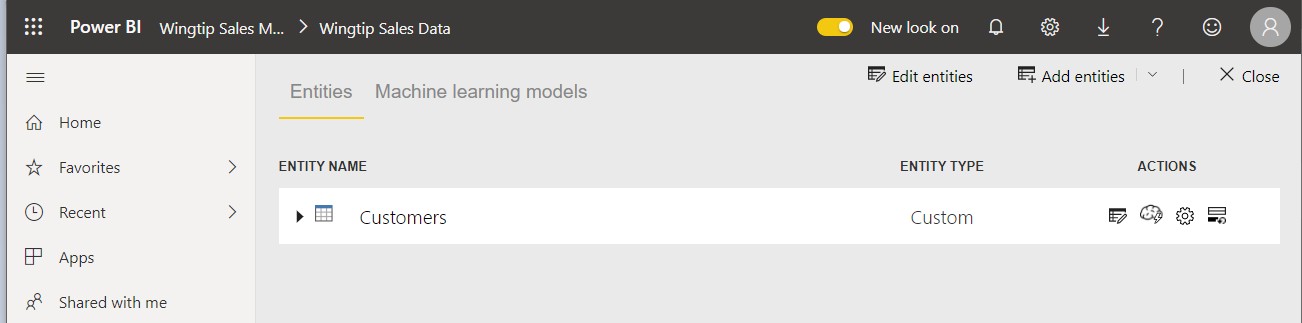
* 1. Drop down the Type menu for the **Customer Type** column and set its value to **Text**.



1. Save the dataflow with a name of **Demo Dataflow**.
   1. Click the **Save & close** button at the bottom right of the **Queries** window.

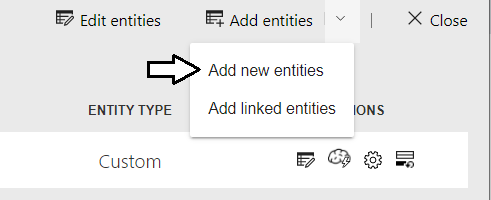


* 1. When prompted by the **Save your dataflow** dialog, enter a name of **Demo Sales Data** and then click **Save**.
  2. You should now see the summary page for the **Demo Sales Data** dataflow.

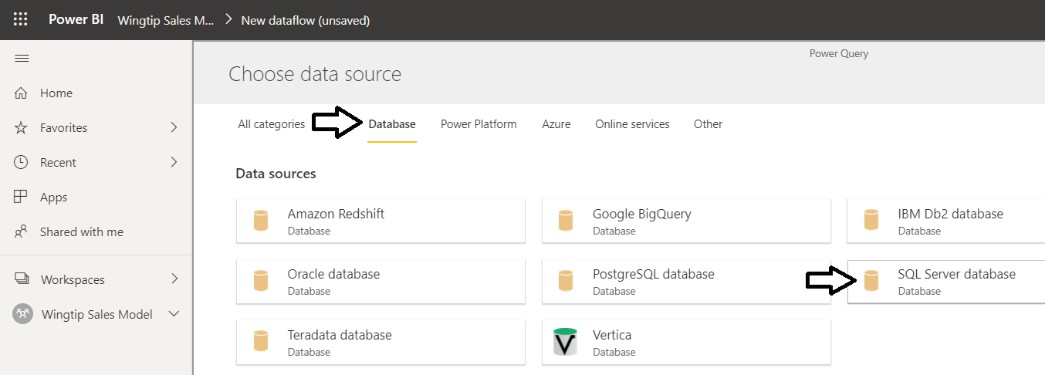


# Exercise 2: Extend the Dataflow by Adding Entities for Products and Sales

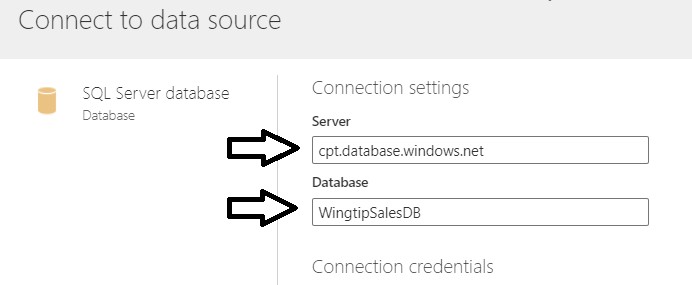
1. Aggiungere delle nuove entità
   1. Locate the **Add entities** menu button in the top right corner of the **Wingtip Sales Data** dataflow summary page.
   2. Drop down the **Add entities** menu button and select the **Add new entities** command.



* 1. On the **Choose data source** page, select the **Database** tab and then select **SQL Server database**.

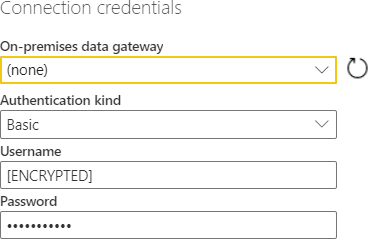


* 1. Enter a **Server** value of **afterhourdemo-ondemand.sql.azuresynapse.net**
  2. Enter a **Database** value of **Demo**.



You should ***NOT*** be required to enter database credentials. That’s because you already entered the credentials for this database in exercise 1 and the credentials are now stored in the Microsoft cloud and will be used transparently.

* 1. Verify that values for **Username:** sqladminuserand **Password:** Pa$$w0rdwere automatically populated when you entered **Server** and **Database**.



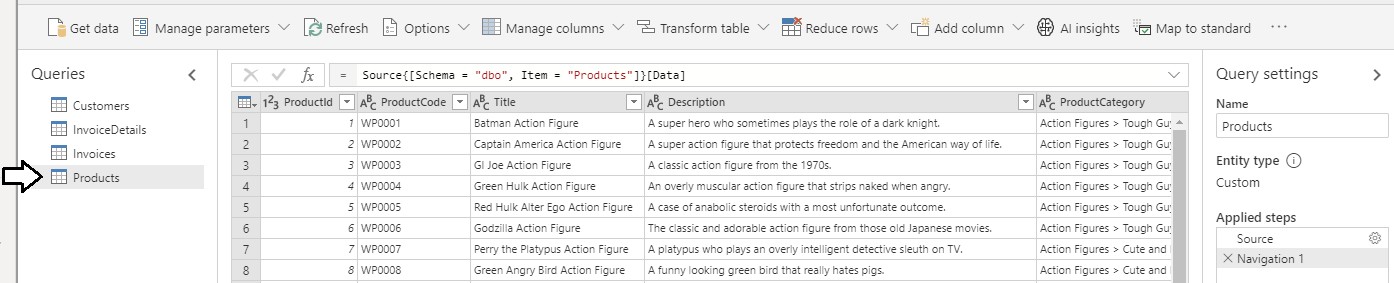
* 1. Click the **Next** button to continue.
  2. When prompted to **Choose data**, select the three tables named **InvoiceDetails** and **Products**.

<https://raw.githubusercontent.com/marcopozzan/Dati/master/PBI_Course_InvoiceDetails.csv>

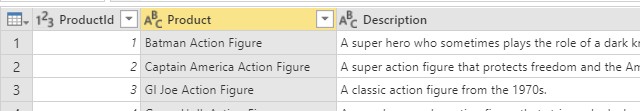
<https://raw.githubusercontent.com/marcopozzan/Dati/master/PBI_Course_Products.csv>

* 1. Click the **OK** button to add three new entity queries to the dataflow.
  2. You should now see three new queries on the **Edit queries** page named **InvoiceDetails**, **Invoices** and **Products**.
  3. Select the **Products** query from the **Queries** list on the left so you can begin to modify its query logic with Power Query.

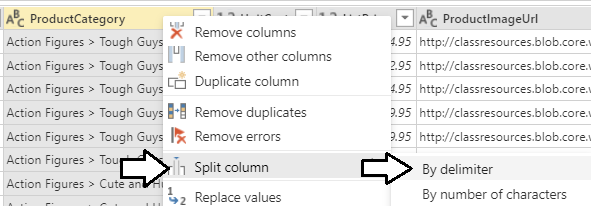
---------------------------------------------------Product -----------------------------------------------



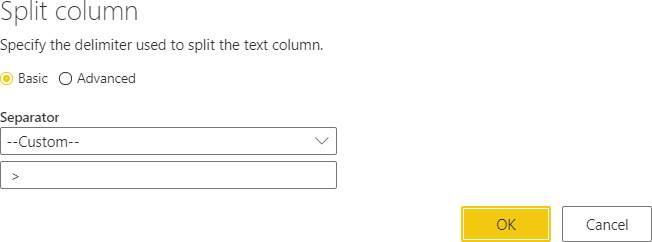
1. Rename the **Title** column to **Product**.
   1. Right-click on the **Title** column and click **Rename**.
   2. Update the column name to **Product**.



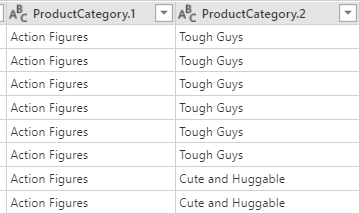
1. Split the **ProductCategory** column up into two separate columns named **Category** and **Subcategory**.
   1. Right-click the **ProductCategory** column and then click the **Split column** > **By delimiter** command.



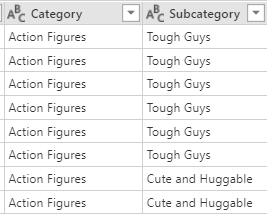
* 1. In the **Split column** dialog, drop down the **Separator** combo box and select **--Custom--**.
  2. In the textbox enter a three-character text value which includes a **space follow** by the **>** character followed by **another space**.
  3. When the **Split column** dialog matches the following screenshot, click the **OK** button.



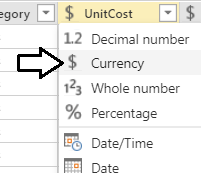
* 1. You should be able to confirm that Power BI Desktop has split the **ProductCategory** column into two separate columns named **ProductCategory.1** and **ProductCategory.2**.



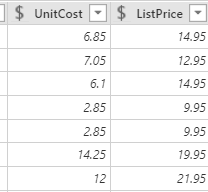
* 1. Rename the **ProductCategory.1** column to **Category** and rename **ProductCategory.2** to **Subcategory**.



1. Modify the column type of the **UnitCost** column and the **ListPrice** column to the **Currency** type.
   1. Use the dropdown column type menu to set the type of the **UnitCost** to **Currency**.



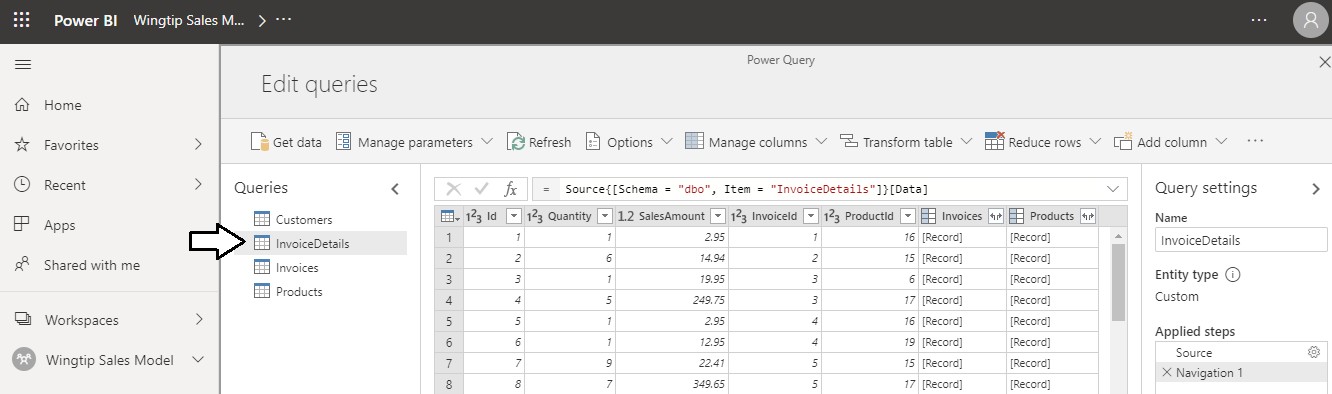
* 1. Use the dropdown column type menu to set the type of the **ListPrice** to **Currency**.



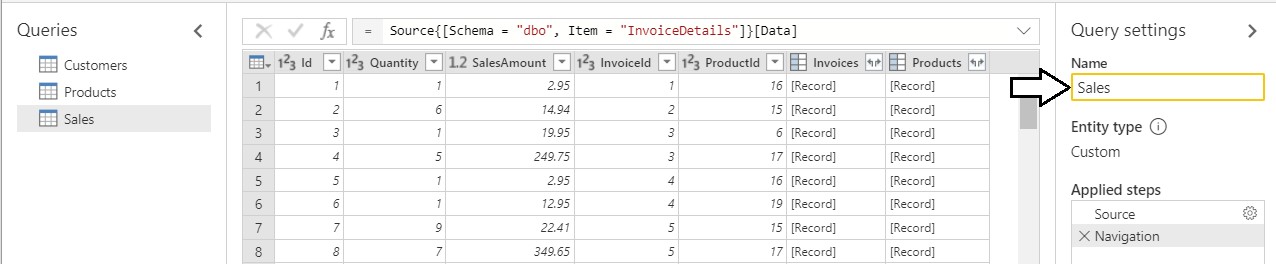
* 1. You have now completed your work on the **Products** query.

---------------------------------------------------Creao SALES DA INVOICES DETAILS -----------------------------------------------

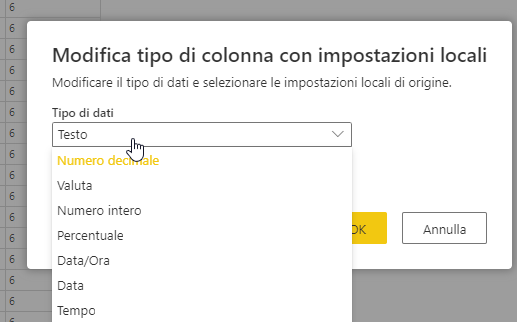
1. Rename the **InvoiceDetails** query to **Sales**.
   1. Select the **InvoiceDetails** query from the **Queries** list on the left.



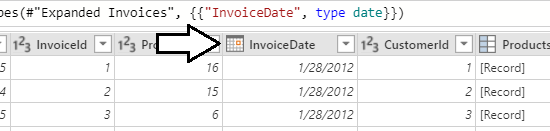
* 1. Update the name of the **InvoiceDetails** query to **Sales** by replacing the text in the **Name** textbox in the **Query Settings** pane.



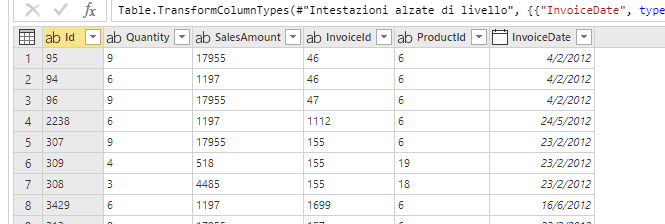
1. Modify the columns of the **Sales** query.
   1. Modify the column type of the **SalesAmount** column to the **Currency** type.



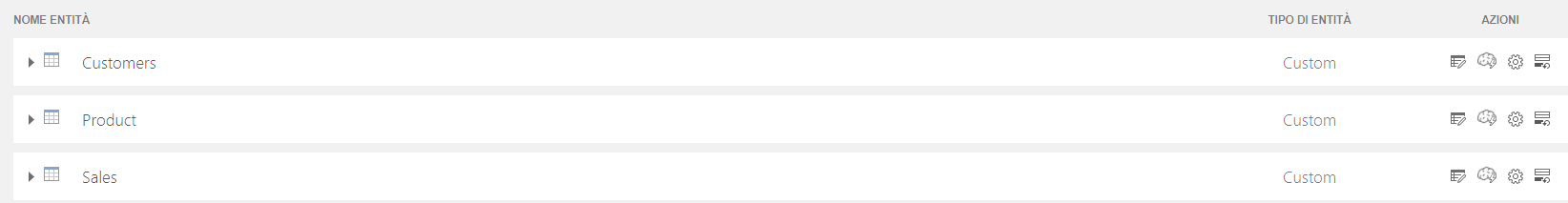
* 1. Modify the column type of the **InvoiceDate** column to the **Date** type.



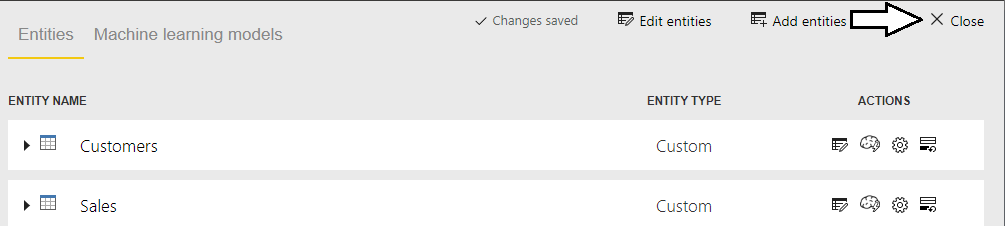
* 1. Modify the column name **id** into **customerid**.



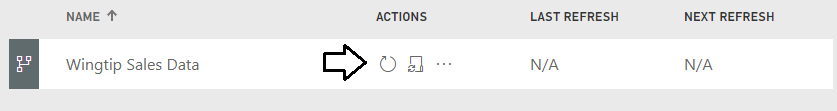
1. Save your work on workflow.
   1. Click the **Save & close** button at the bottom right of the page to save your work and return to the dataflow summary page.
   2. The **Entities** list should now display four entities named **Customers**. **Sales**, and **Products**.



1. Refresh the dataflow to populate it with data.
   1. Click the **Close** button in the upper right corner of the dataflow summary page.



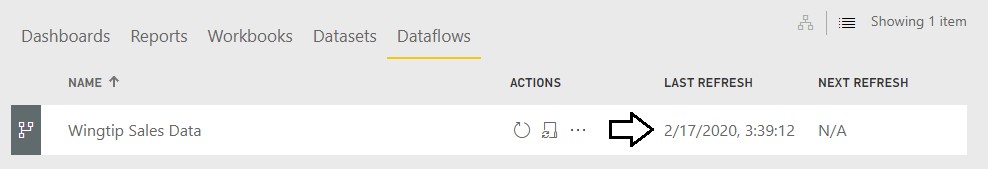
* 1. You should now see the **Wingtip Sales Data** dataflow in the app workspace summary page.
  2. Click the Refresh button to begin a refresh operation on the **Wingtip Sales Data** dataflow.



* 1. Wait for the refresh operation to complete. It might take one or two minutes to complete.



* 1. Once the refresh operation completes, you should see the **LAST REFRESH** time has been updated.



# Exercise 3: Importing Dataflow Entity Data with Power BI Desktop

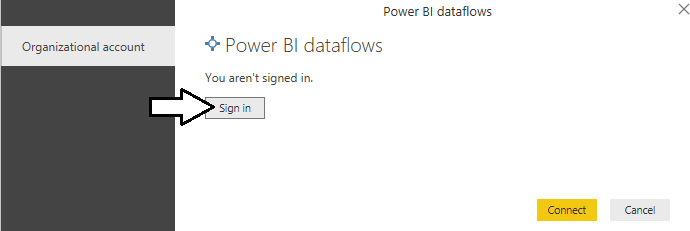
1. Launch Power BI Desktop to start a new project.
2. Save the new project as **Dataflow.pbix** using the following path.

**C:\Student\Projects\Dataflow.pbix**

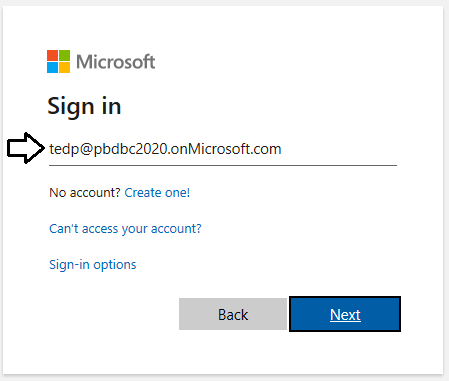
1. Import all four entities from the **Wingtip sales Data** dataflow.
   1. Drop down the **Get Data** menu button on the ribbon and click **Power BI dataflows se non uso un data lake custom ma gestito da Power BI . Altrimenti Power Platform dataflow**.



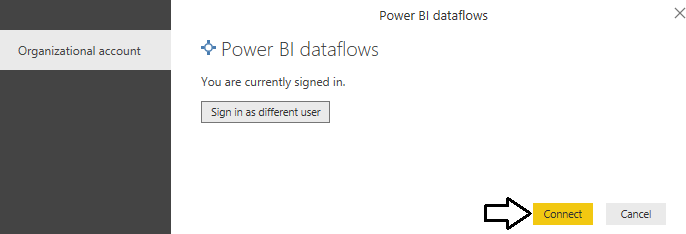
* 1. When prompted by the **Power BI dataflows** connect dialog, click **Sign in**.



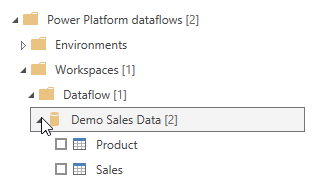
* 1. In the **Sign in** dialog, enter your account name and then click **Next**.



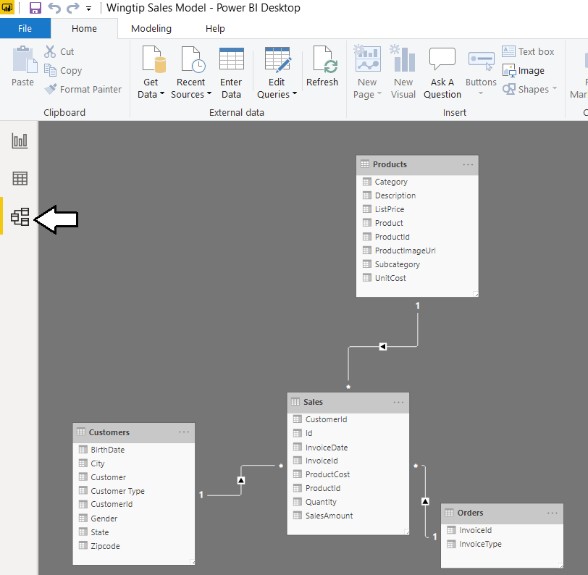
* 1. Once you have signed in, click the **Connect** button.



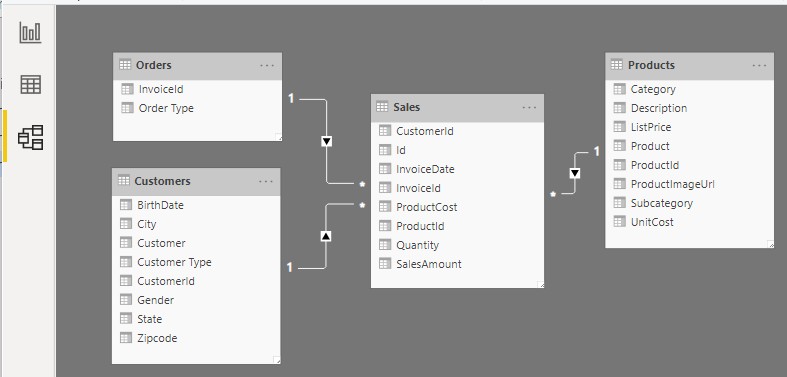
* 1. In the **Navigator** dialog, expand the **Wingtip Sales Model** workspace
  2. Expand the **Wingtip Sales Data** workflow so you can see the entities inside this dataflow.



* 1. Once the queries execute, click the **Model View** button to see the four tables imported from the dataflow.



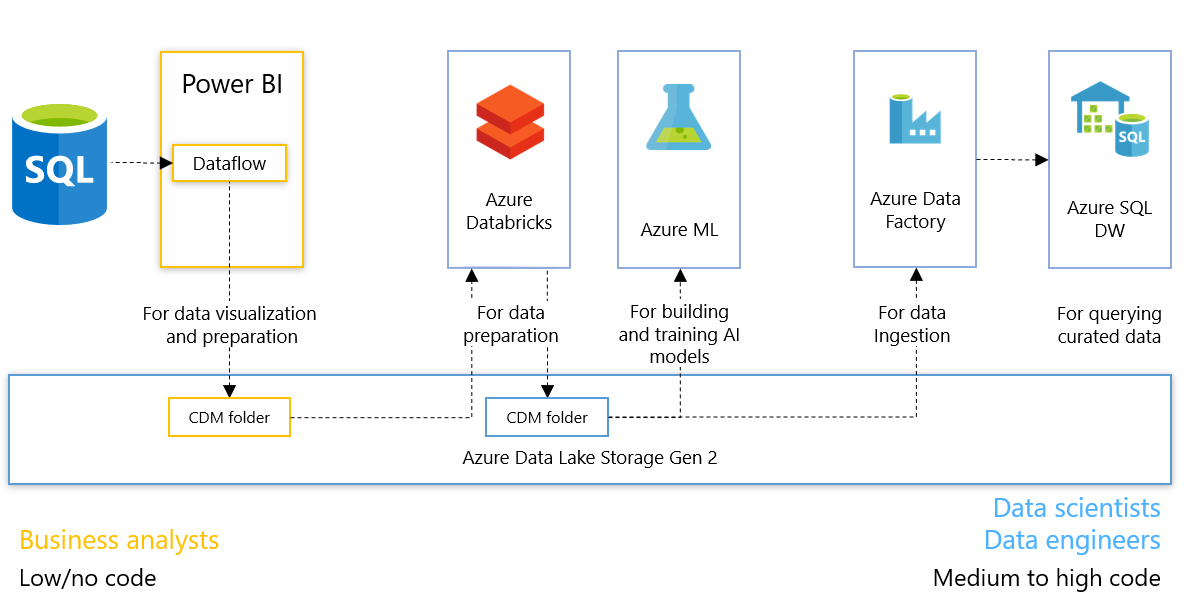
* 1. Using the mouse, rearrange the four tables in Model View to match the following screenshot.



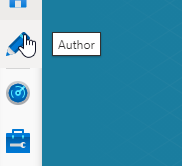
* 1. Save your work to **dataflow.pbix**.

Demo 3

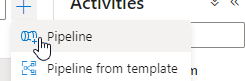
Fare un progetto per pescare da adf (vedi articolo perico)



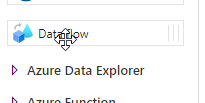
Aprire un progetto di Synapse pipeline



Creiamo una pipeline



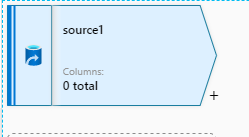
Scegliamo il dataflow



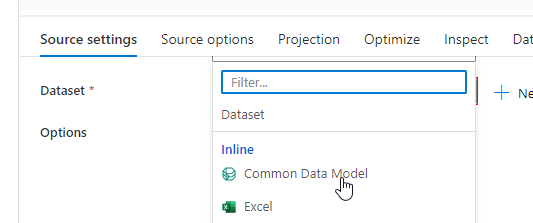
Poi indichiamo il data flow



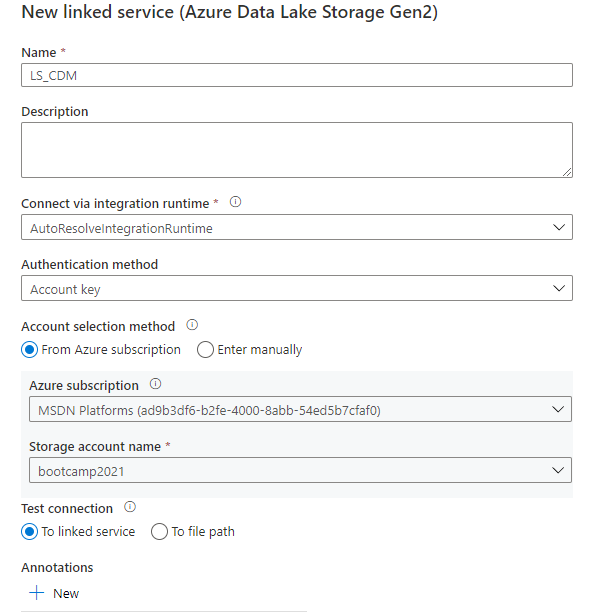
Clicchiamo su source



Nella source scegliamo **Common** **Data Model**

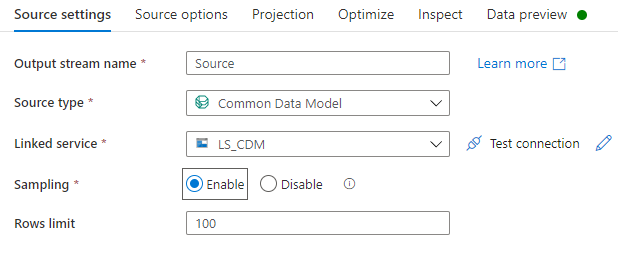


Selezioniamo il **linked service** e ne creiamo uno nuovo

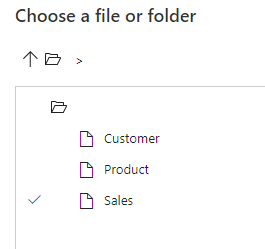
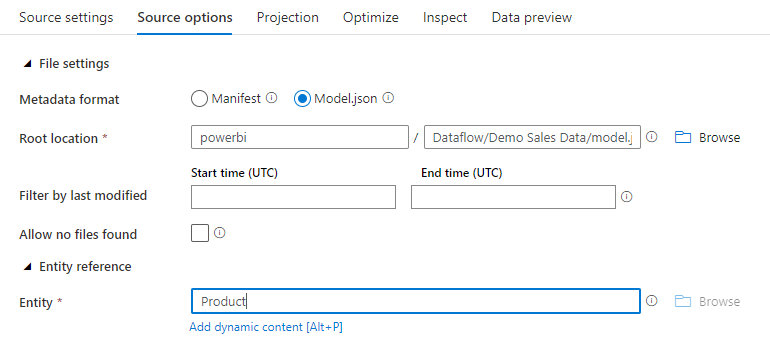


Prima di continuare attivare il Data flow debug!!!

Definiamo gli altri parametri su **Source setting**



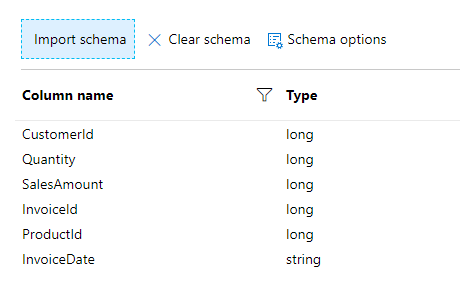
Specificare su **source option** il valore di root location **Model.json** indicare l’entity **Product**



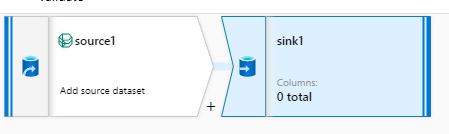
Ora nel blade Projection bisogna importare lo schema



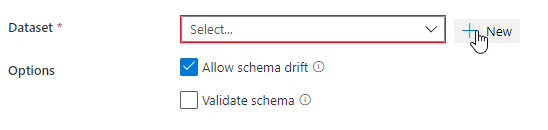
Schema importato



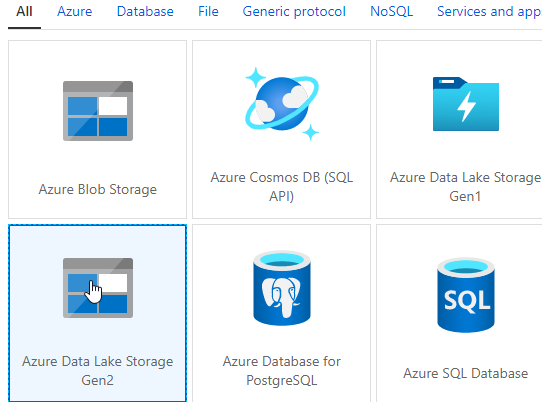
Agganciare il sink per la destinazione



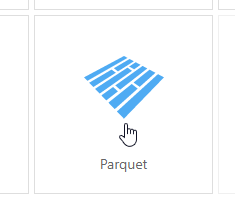
Nelle proprietà del sink aggiungiamo un dataset



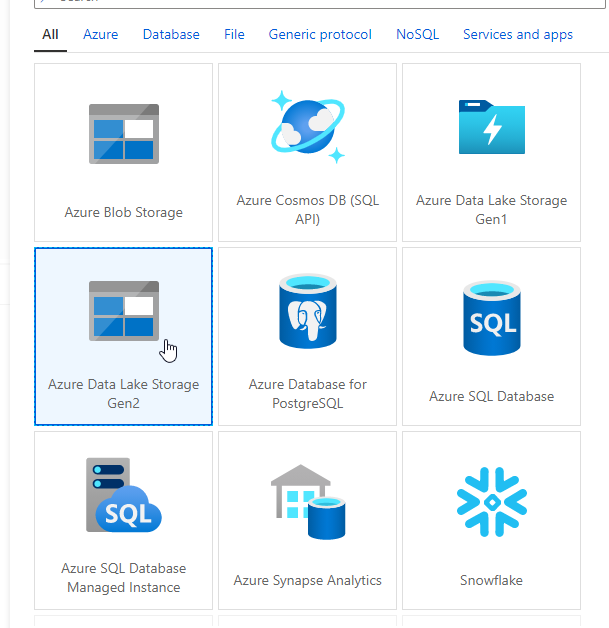
Scegliamo che possa scrivere in un ADLS Gen2



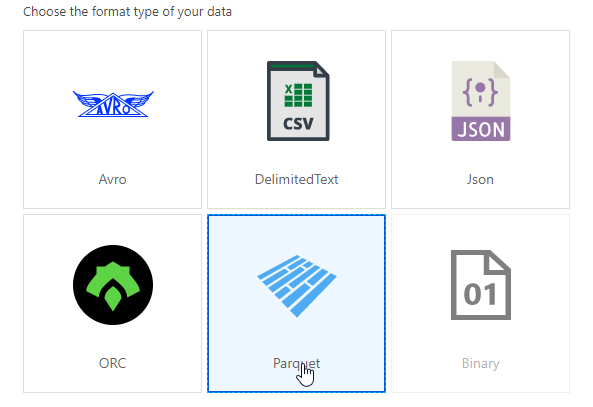
E poi scegliamo un file parquet



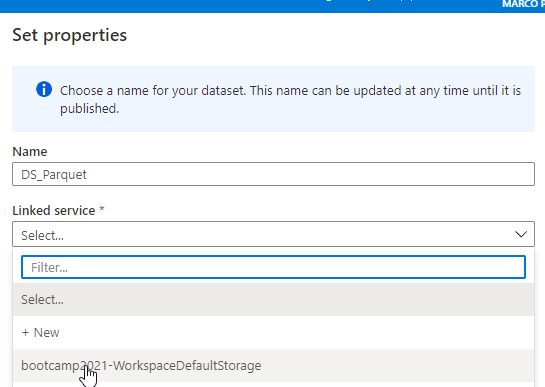
Inserire un nuovo dataset che deve essere Azure data lake storage gen2



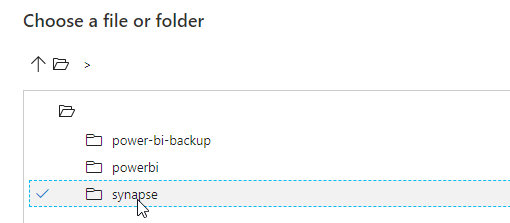
Selezioniamo un file parquet



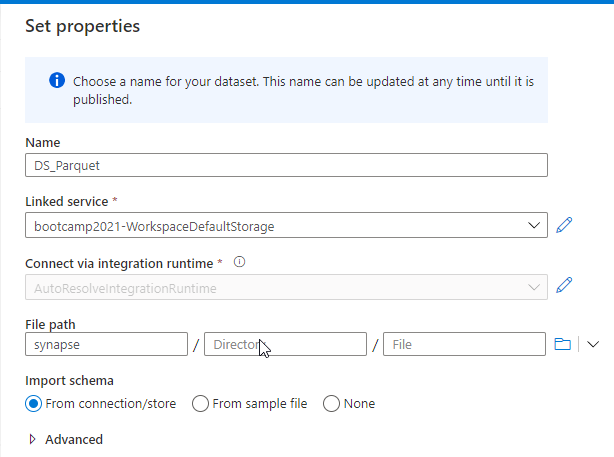
Selezioniamo un file parquet e poi andiamo a definire il nome del dataset **DS\_parquet e il linked service che è quello di synapse**



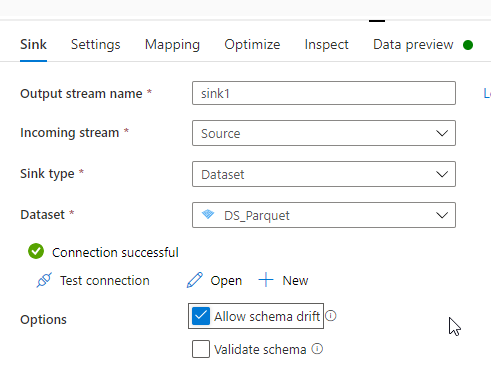
il file path impostato su **synapse**

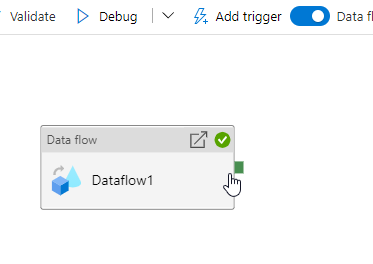
****

**La configurazione finale del Dataset sarà questa!!!!!!!!!!!!!!!!!!!!!!!**

****

**La configurazione finale del Sink invece sarà questa**





**Scriviamo una query di test**

SELECT

    TOP 100 \*

FROM

    OPENROWSET(

        BULK 'https://bootcamp2021.dfs.core.windows.net/synapse/part-00000-7e2d1d4e-9645-4905-8af4-8cc9f1a2030d-c000.snappy.parquet',

        FORMAT='PARQUET'

    ) AS [result]

**Creiamo un database**

CREATE DATABASE bootcamp2001

**Creiamo tabella esterna**

IF NOT EXISTS (SELECT \* FROM sys.external\_file\_formats WHERE name = 'SynapseParquetFormat')

    CREATE EXTERNAL FILE FORMAT [SynapseParquetFormat]

    WITH ( FORMAT\_TYPE = PARQUET)

GO

IF NOT EXISTS (SELECT \* FROM sys.external\_data\_sources WHERE name = 'synapse\_bootcamp2021\_dfs\_core\_windows\_net')

    CREATE EXTERNAL DATA SOURCE [synapse\_bootcamp2021\_dfs\_core\_windows\_net]

    WITH (

        LOCATION   = 'https://bootcamp2021.dfs.core.windows.net/synapse',

    )

Go

CREATE EXTERNAL TABLE ext\_sales (

    [CustomerId] bigint,

    [Quantity] bigint,

    [SalesAmount] bigint,

    [InvoiceId] bigint,

    [ProductId] bigint,

    [InvoiceDate] varchar(8000)

    )

    WITH (

    LOCATION = 'part-00000-7e2d1d4e-9645-4905-8af4-8cc9f1a2030d-c000.snappy.parquet',

    DATA\_SOURCE = [synapse\_bootcamp2021\_dfs\_core\_windows\_net],

    FILE\_FORMAT = [SynapseParquetFormat]

    )

GO

SELECT TOP 100 \* FROM ext\_sales

GO