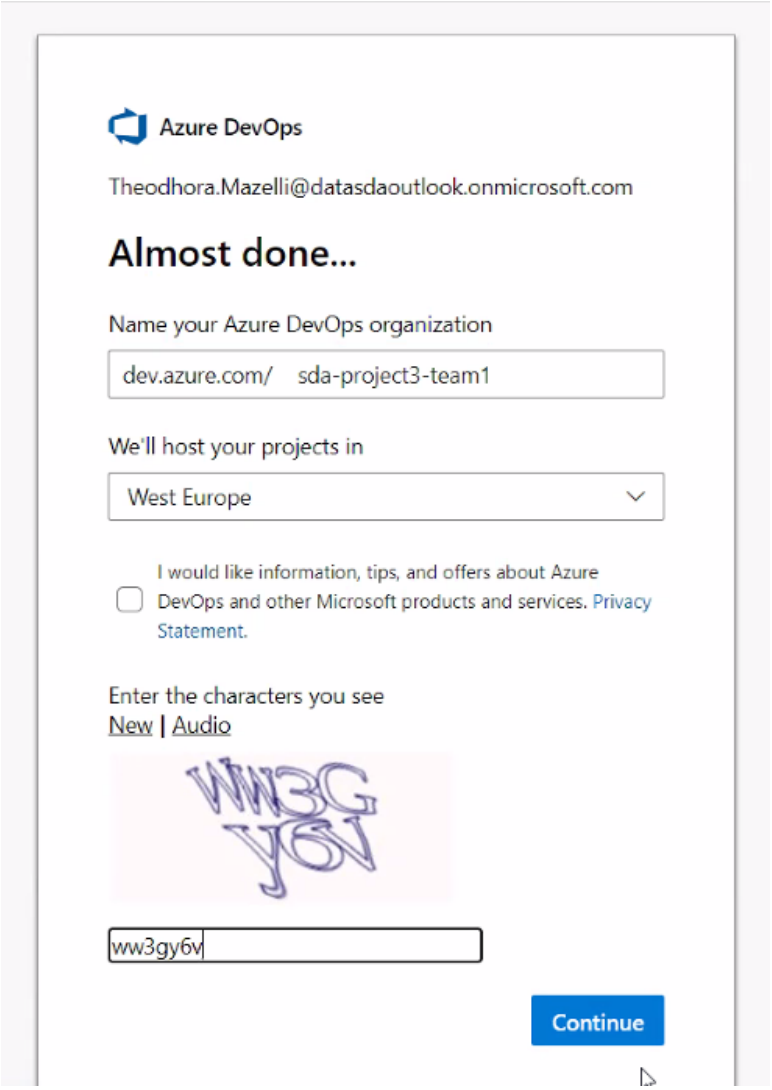
Azure Data Science Project

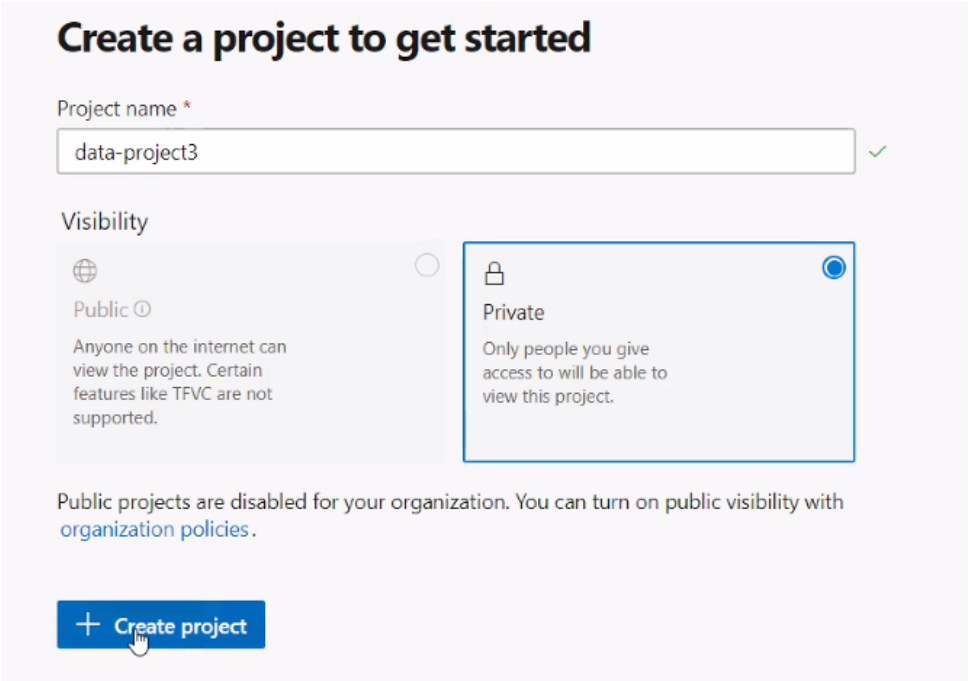
Solution:
Connect to SQL Server instance using these credentials:
Server Name: rg-data-project.database.windows.net
Login: sqladminuser
Password: Admin123+

For this project we will use Azure Data Factory, Azure Portal, Azure Devops Portal and SQL Server and Streamlit.

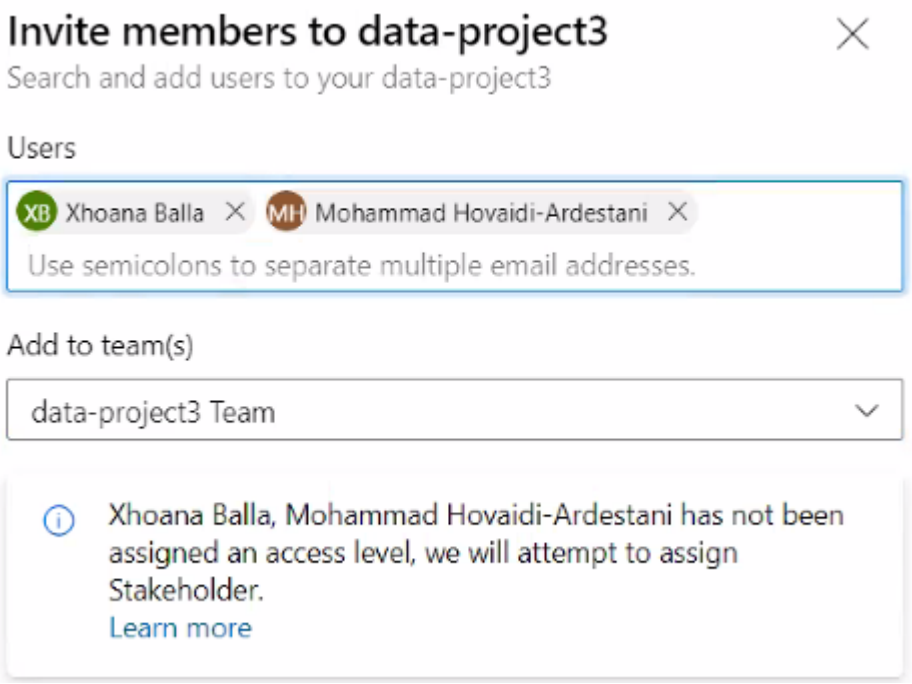
In Azure DevOps let's create a new organisation:



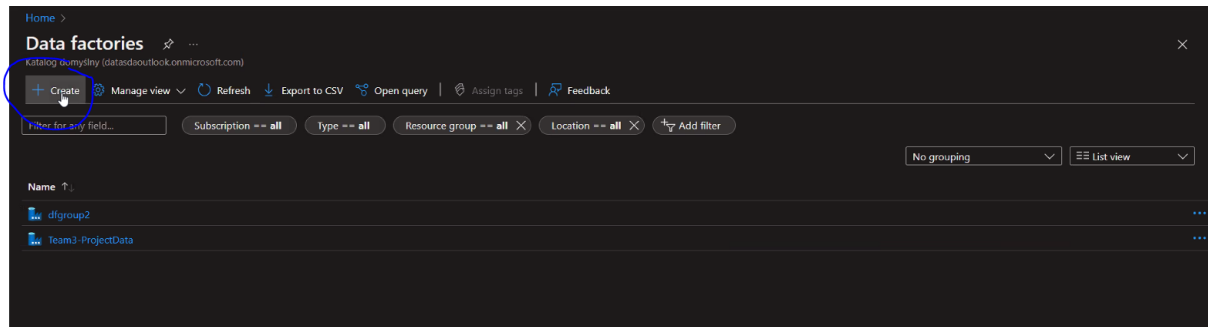
After the organisation is created we should create a new project:



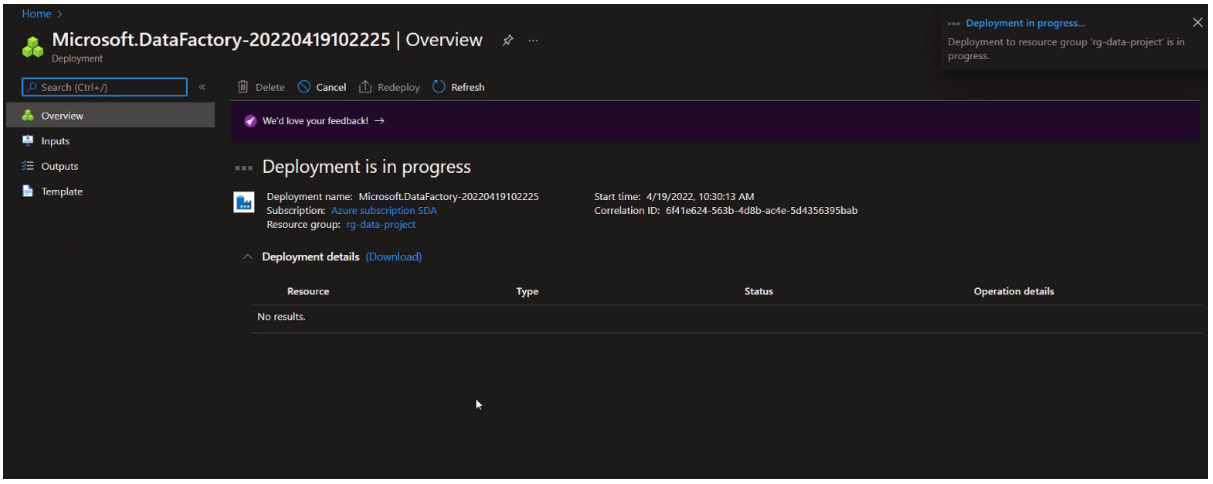
Add the members in the project:



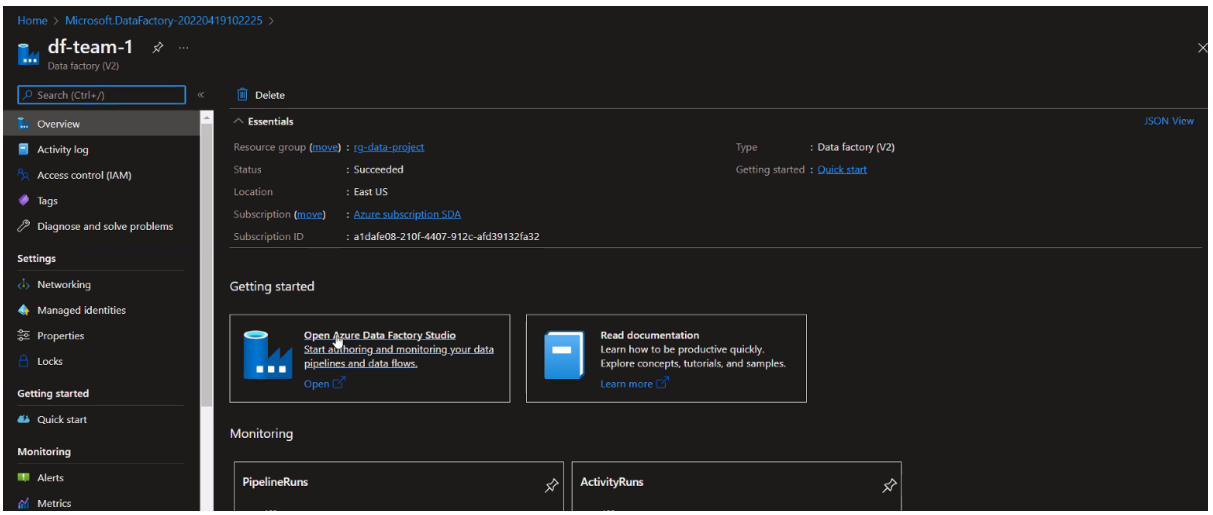
In the azure portal let’s create a new datafactory:



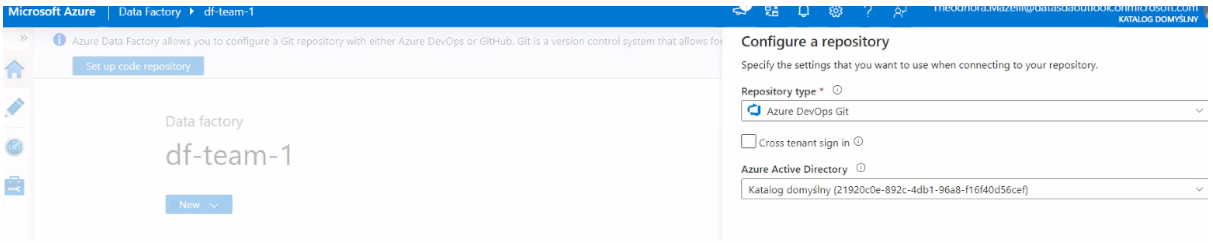
Fill in the information for the creation of the data factory.



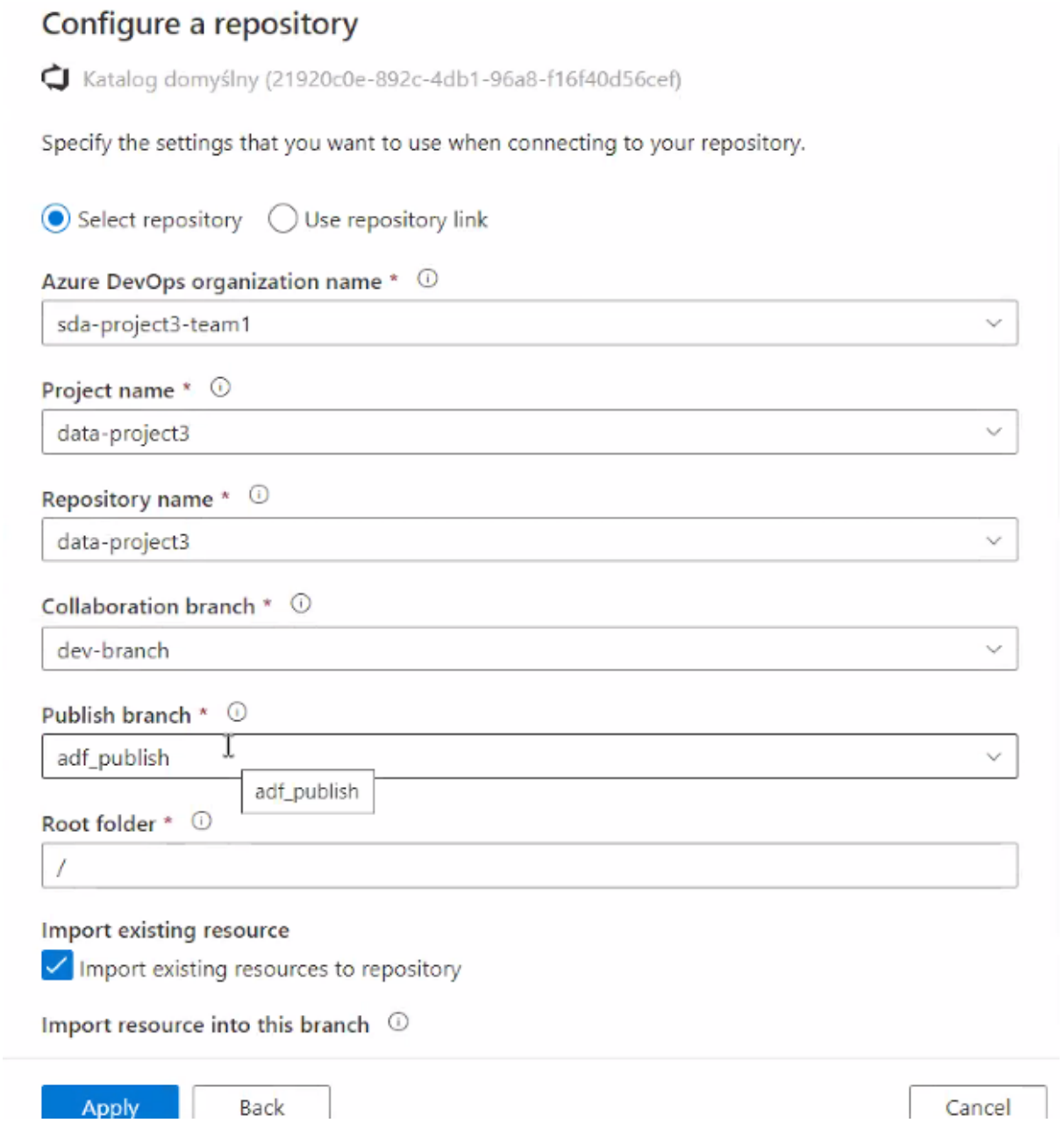
The data factory is created:

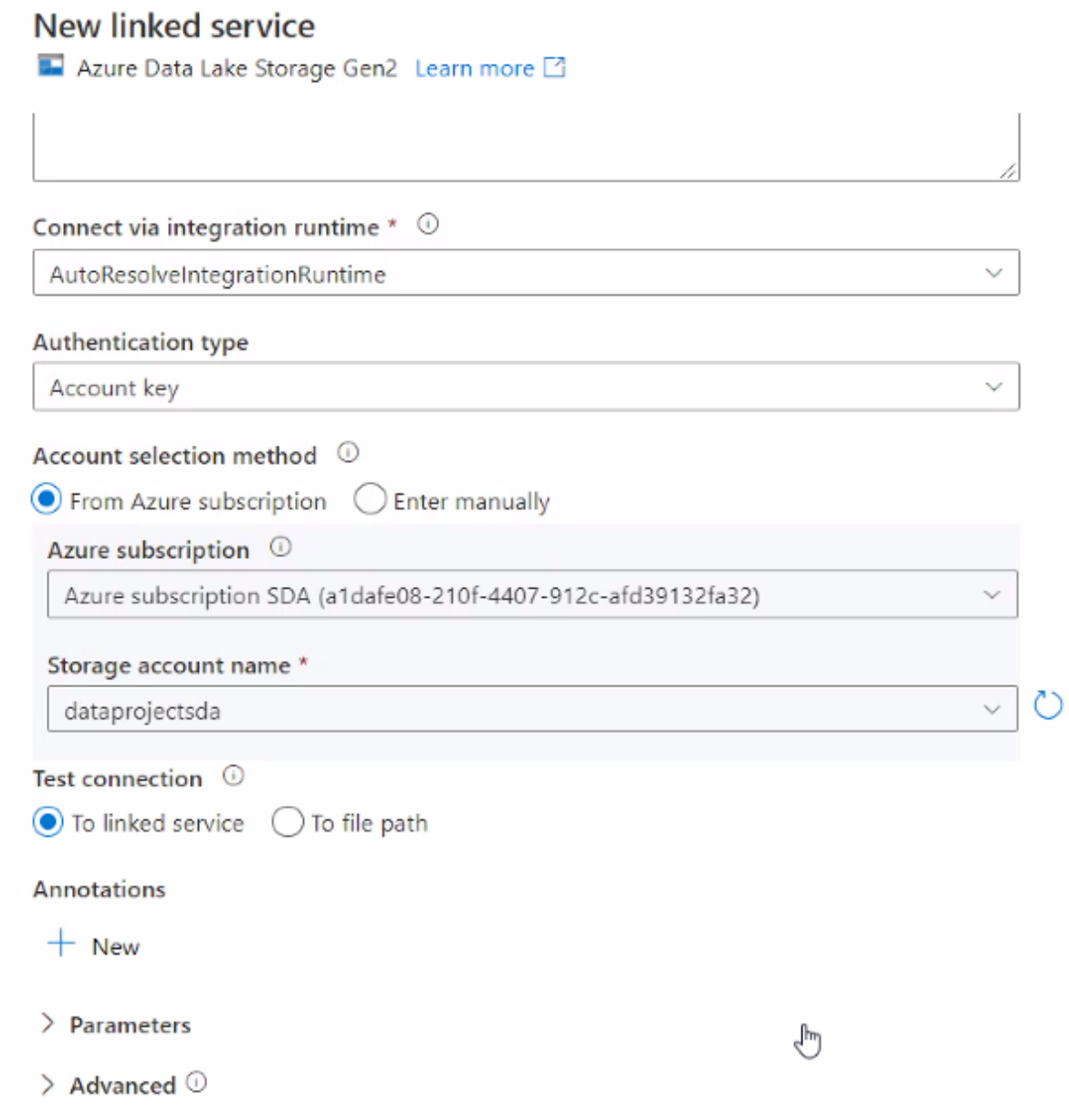


We connect the datafactory to the project we have created in devOpd portal:

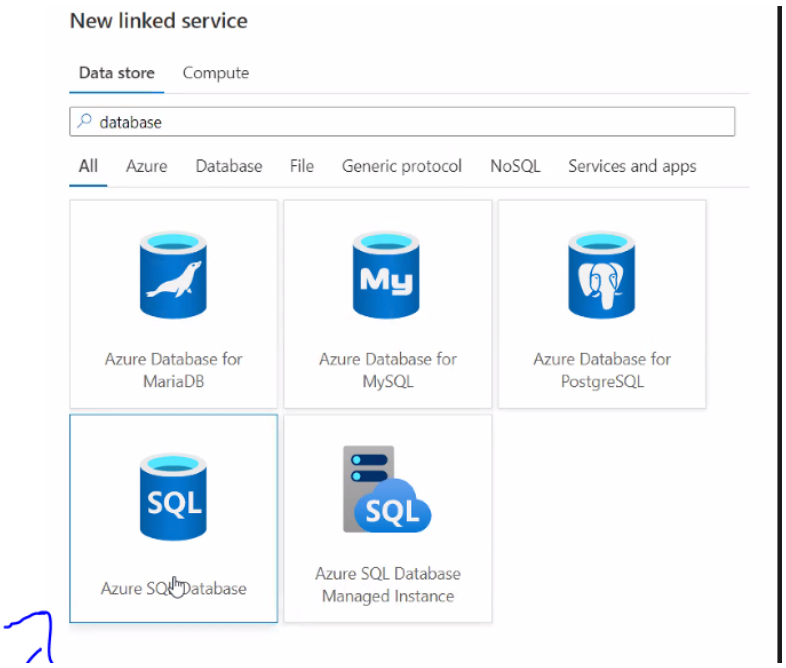


For configuring the repository we fill these information;





Let’s create linked service with sql:



New linked service

Azure SQL Database Learn more

Connect via integration runtime *

AutoResolveIntegrationRuntime

Connection string | Azure Key Vault

Account selection method

From Azure subscription | Enter manually

Azure subscription

Azure subscription SDA (a1dafe08-210f-4407-912c-afd39132fa32)

Server name *

rg-data-project

Database name *

rg-data-project-team1

Authentication type *

SQL authentication

User name *

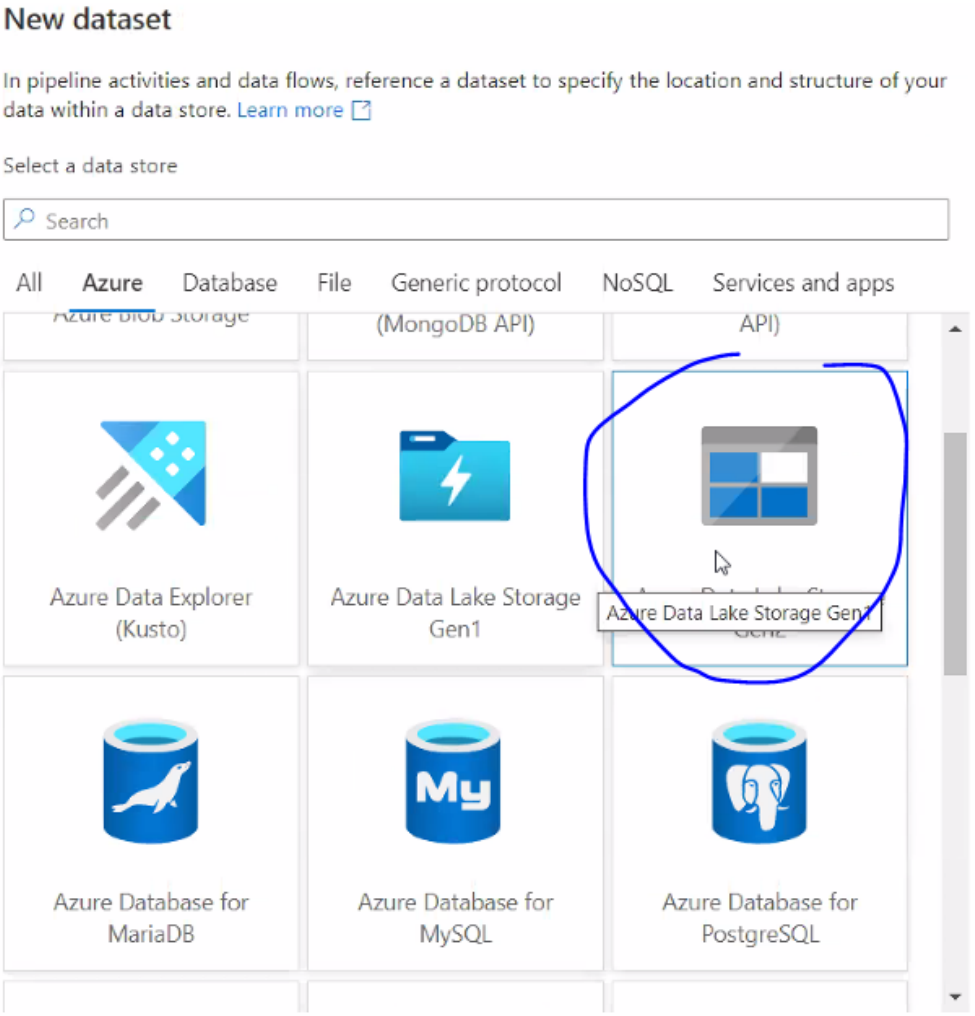
sqladminuser

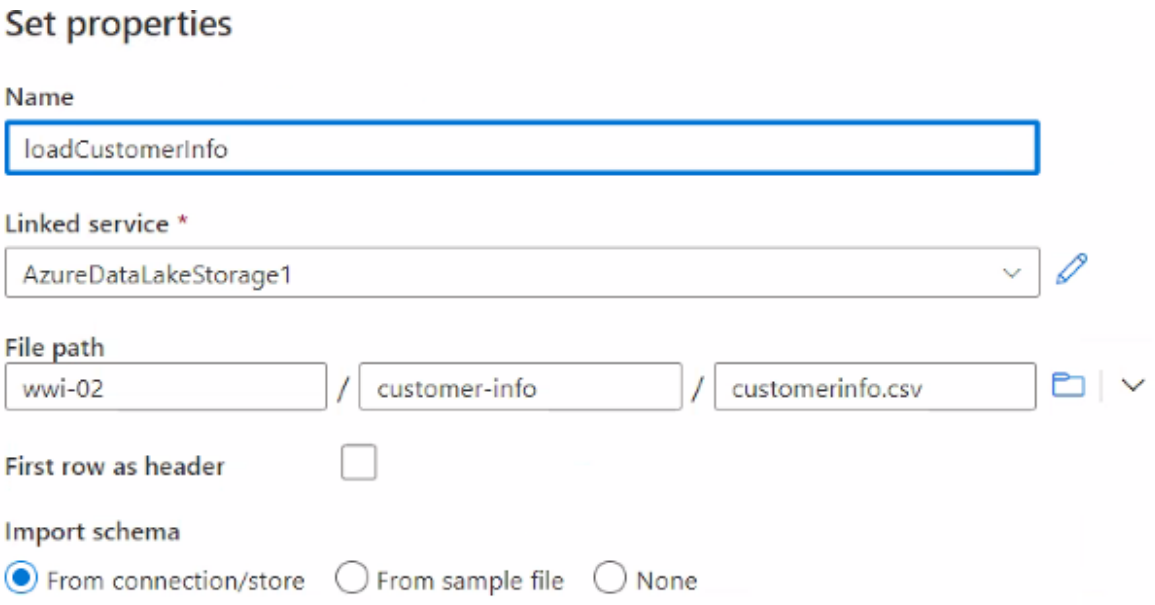
Add dynamic content [Alt+Shift+D]

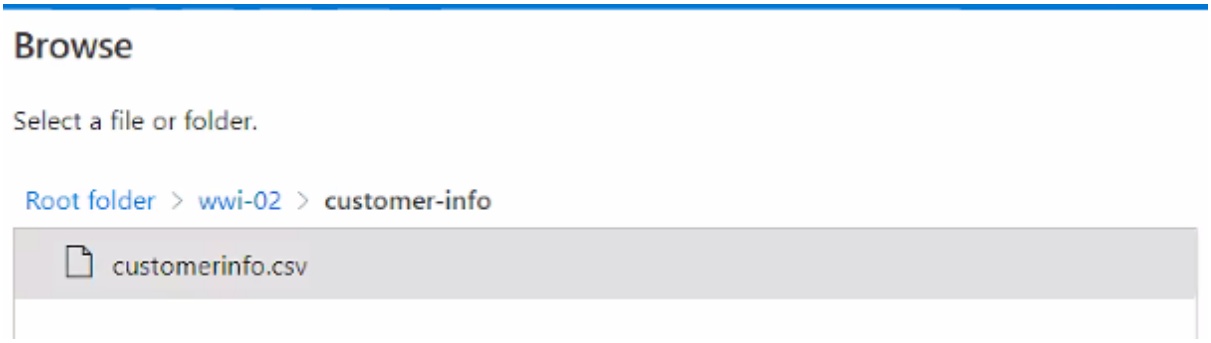
Password | Azure Key Vault

Password *

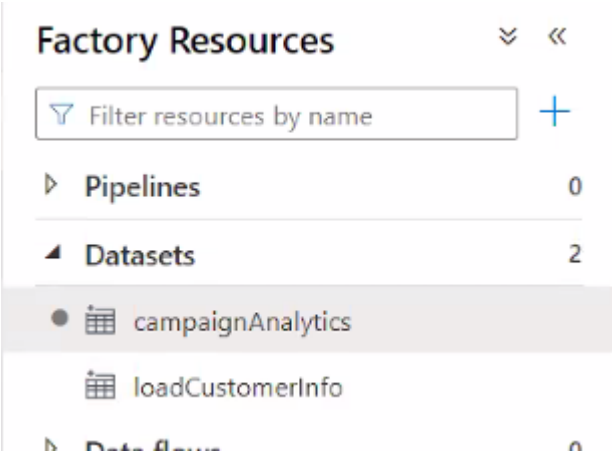
Create import a dataset:



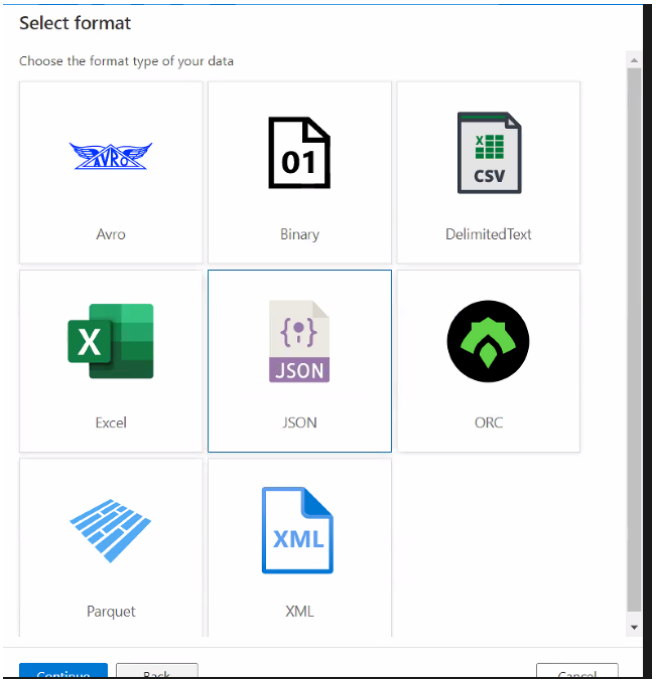


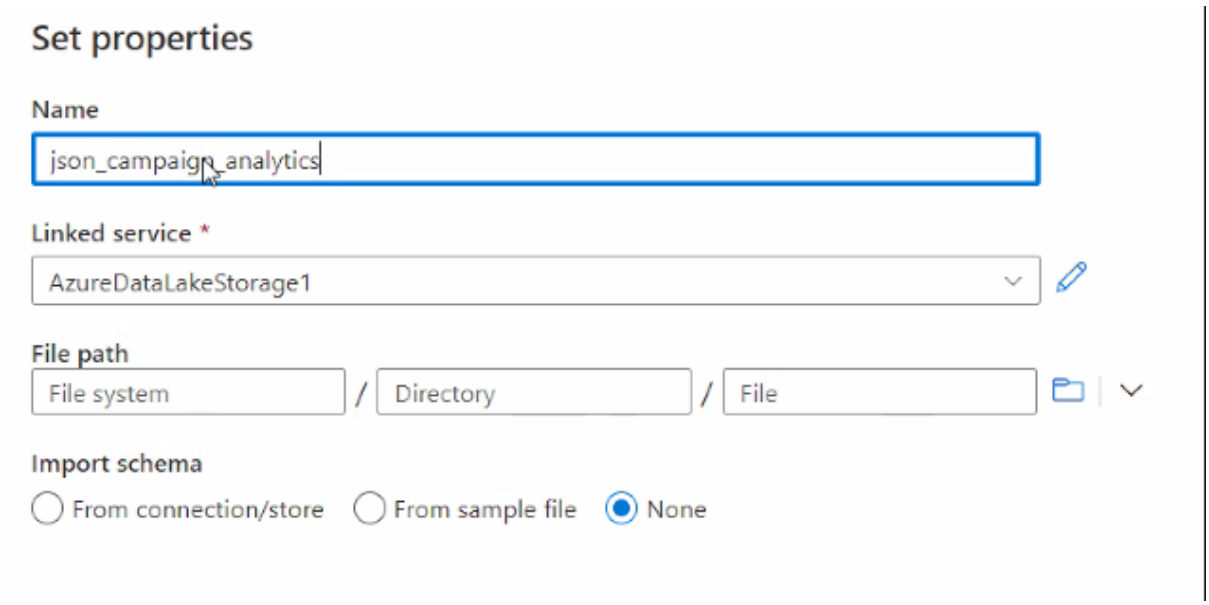


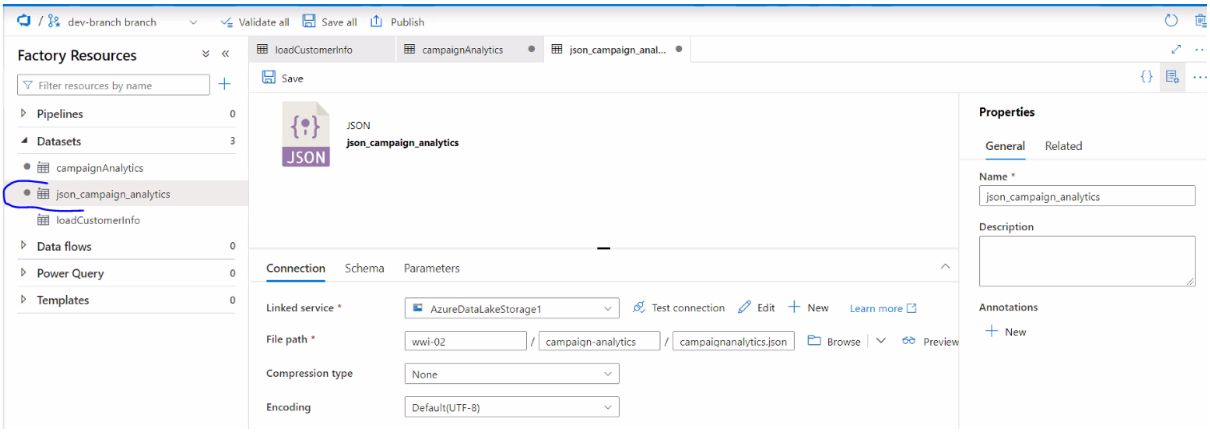
Import another csv file:



Now we will import the json file:

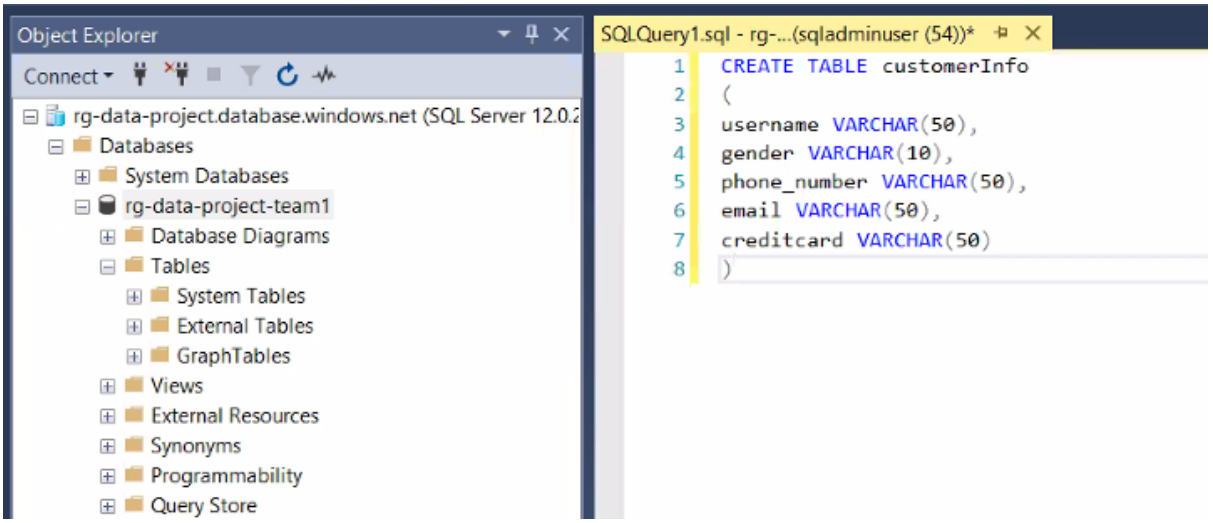




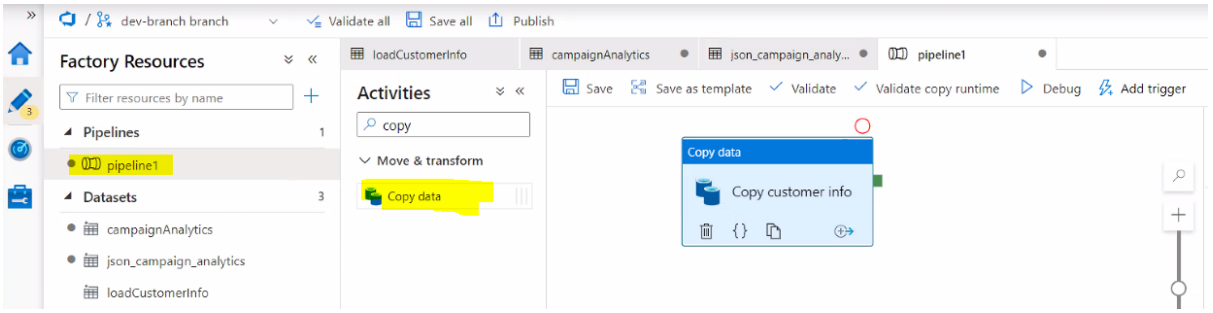


Now we will create the pipelines for displaying the data we have into the SQL server. For doing this we will create the tables in our database in SQL Server.

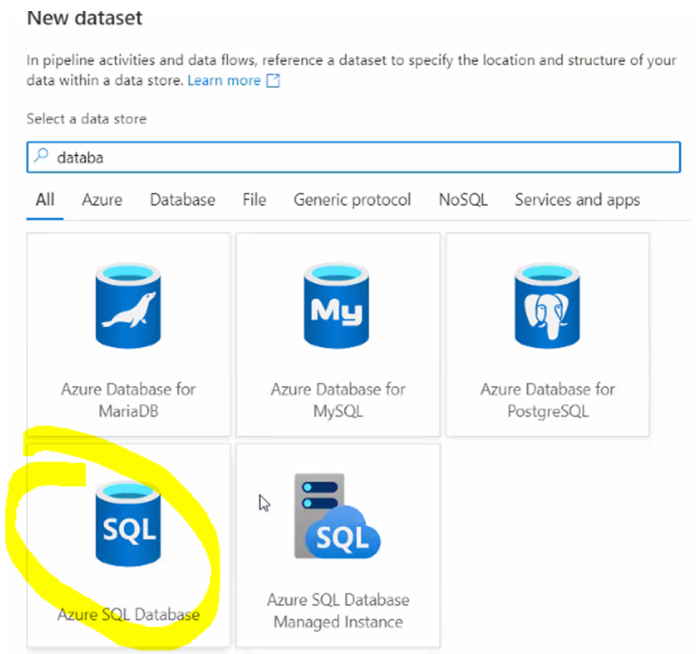
```
CREATE TABLE customerInfo
(
username VARCHAR(50),
gender VARCHAR(10),
phone_number VARCHAR(50),
email VARCHAR(50),
creditcard VARCHAR(50)
)
```

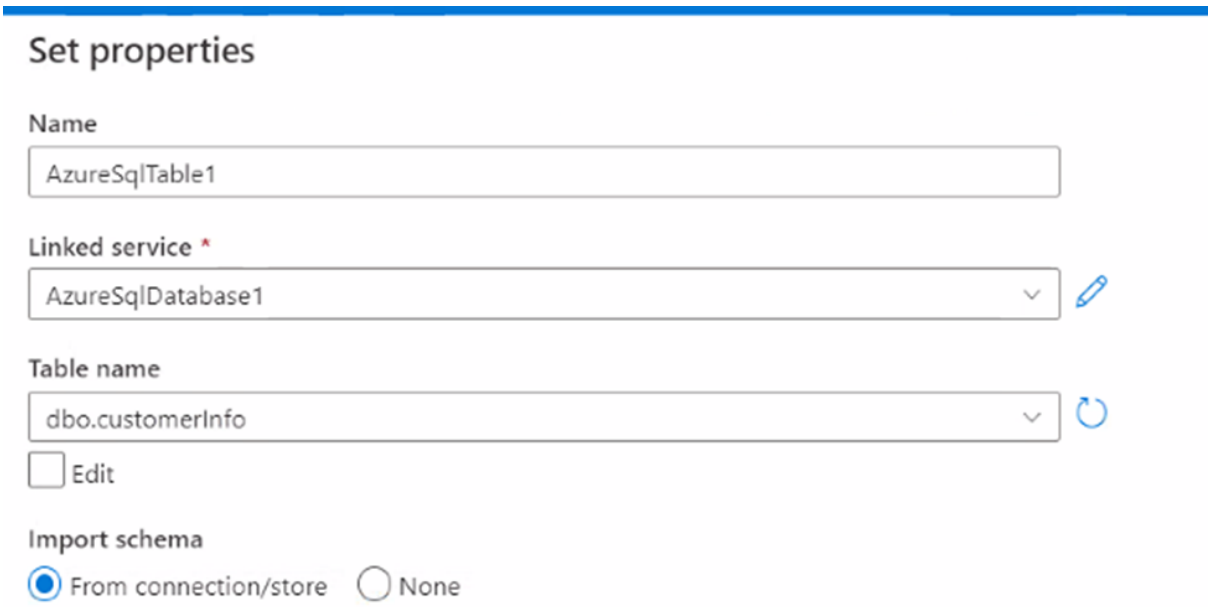


In Azure Data Factory let's create a pipeline to load the data:

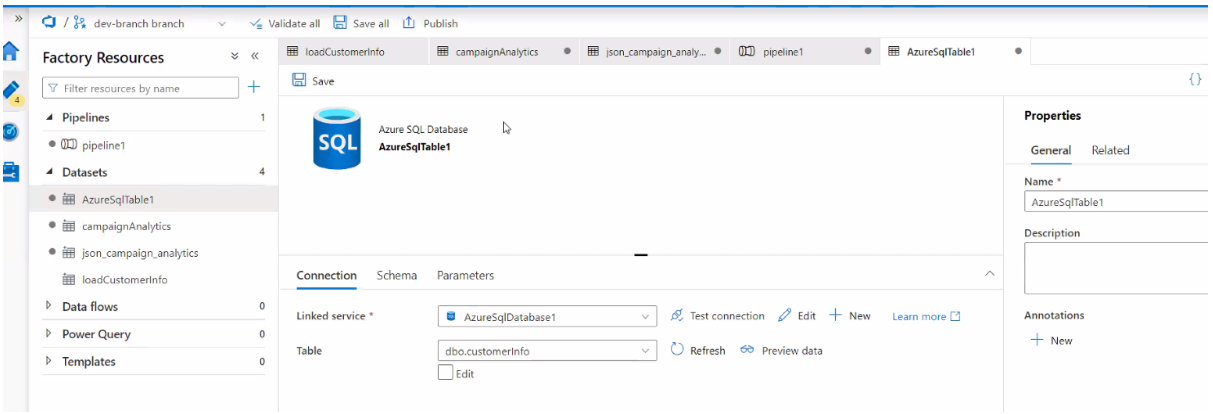


We create a dataset for loading the data:



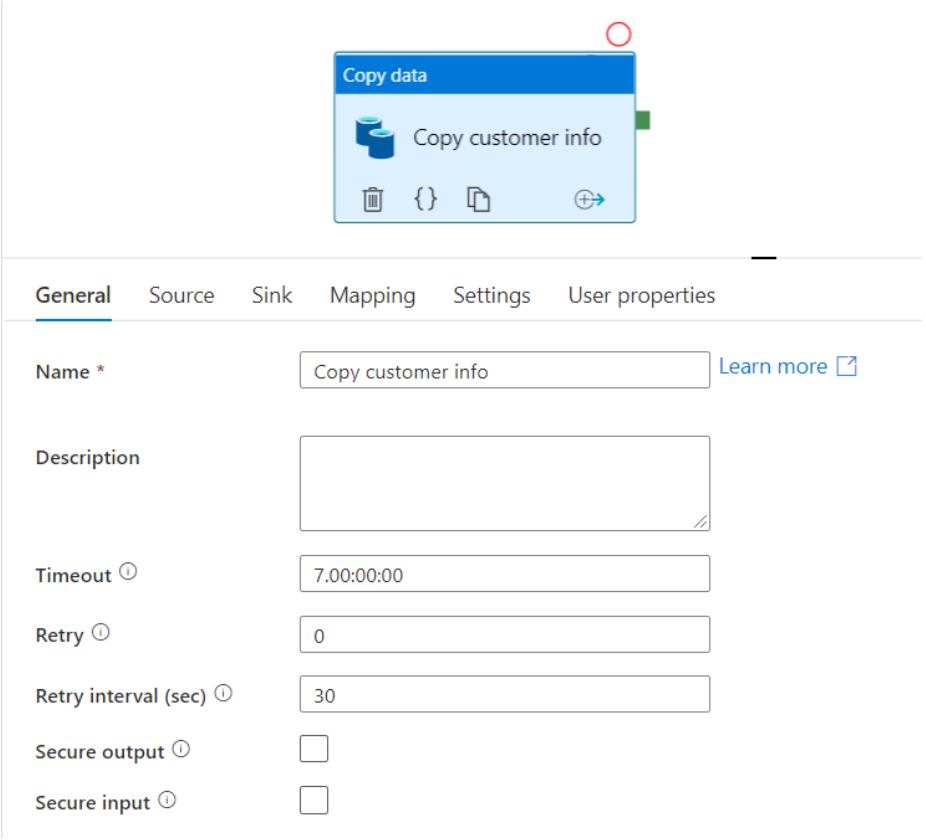


The file is created:

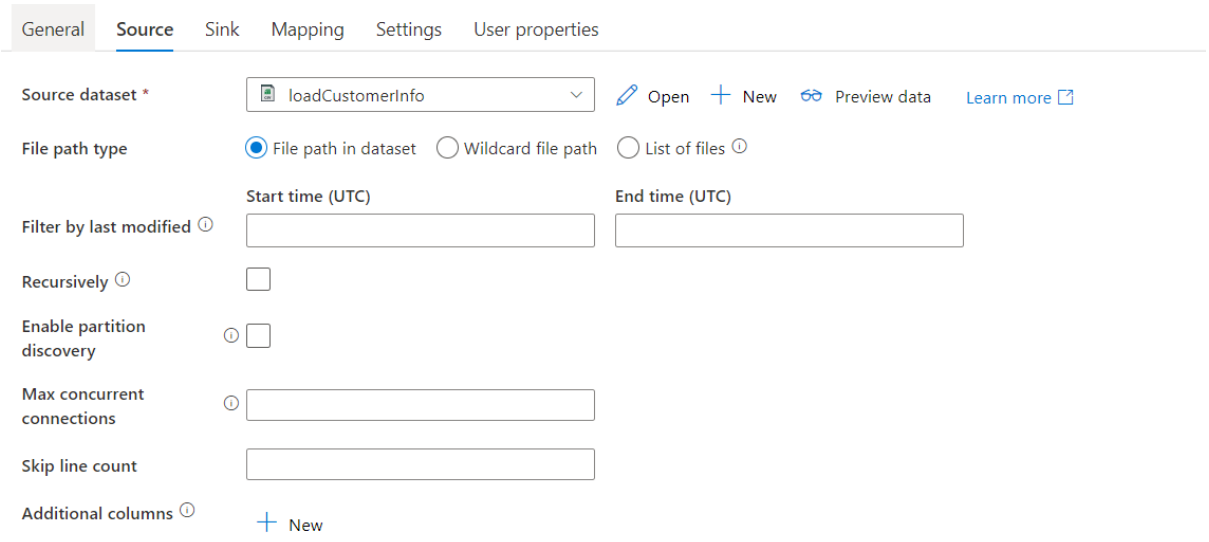


Now let’s configure the pipeline:

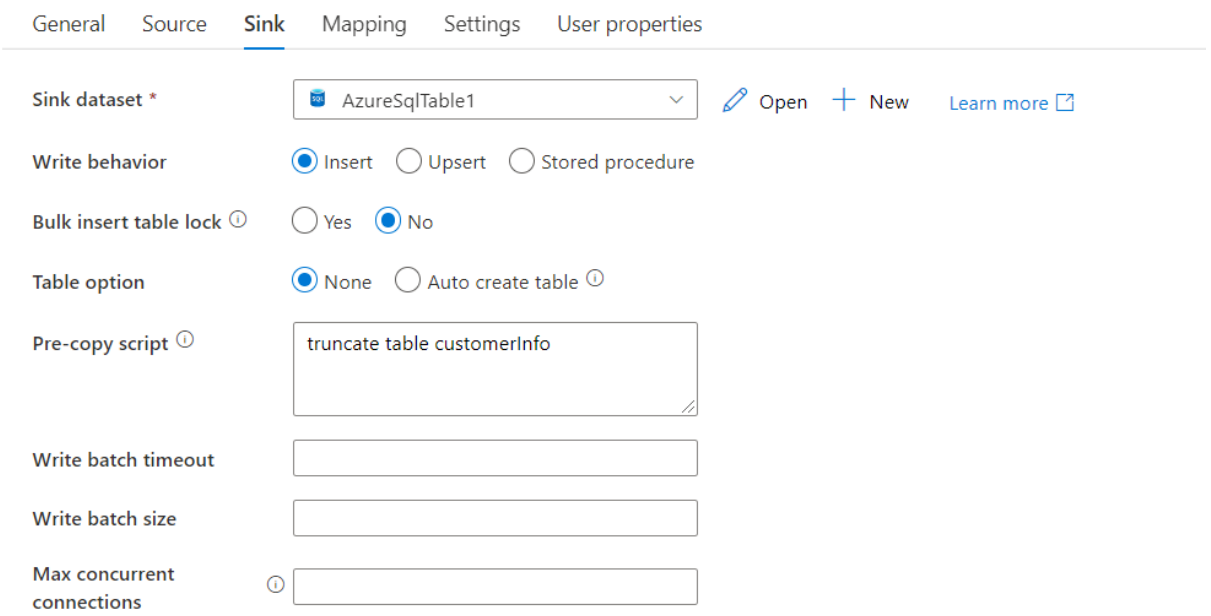
See below:


Copy data
Copy customer info
General
Source
Sink
Mapping
Settings
User properties
Name *
Copy customer info
Learn more
Description
Timeout
7.00:00:00
Retry
0
Retry interval (sec)
30
Secure output
Secure input

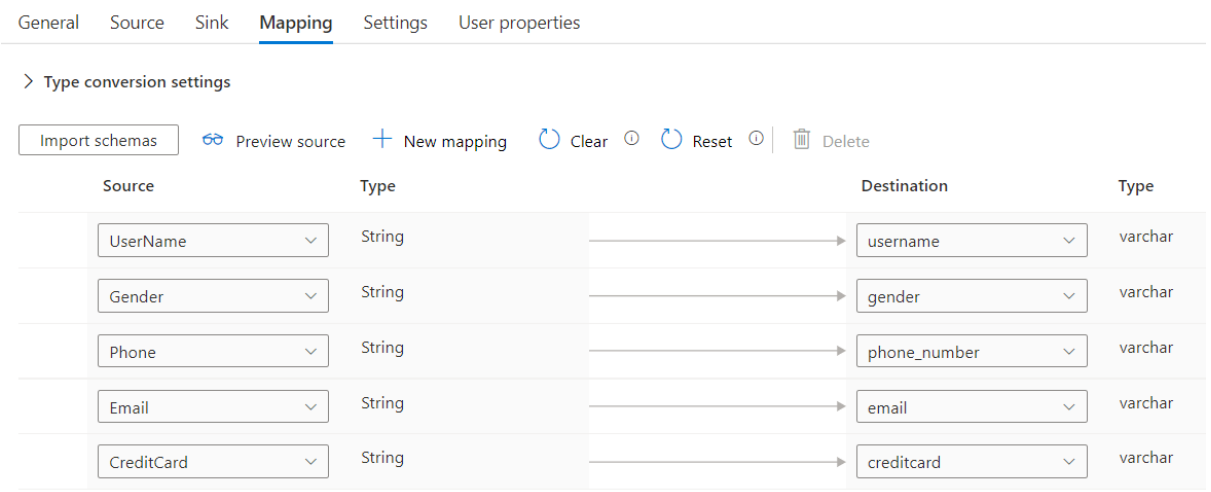
Source:


General
Source
Sink
Mapping
Settings
User properties
Source dataset *
loadCustomerInfo
Open
New
Preview data
Learn more
File path type
File path in dataset
Wildcard file path
List of files
Start time (UTC)
End time (UTC)
Filter by last modified
Recursively
Enable partition discovery
Max concurrent connections
Skip line count
Additional columns
New

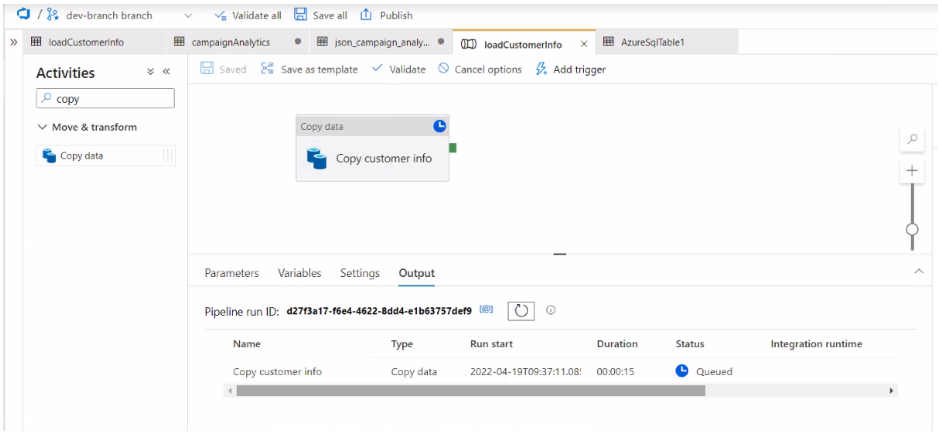
Sink:

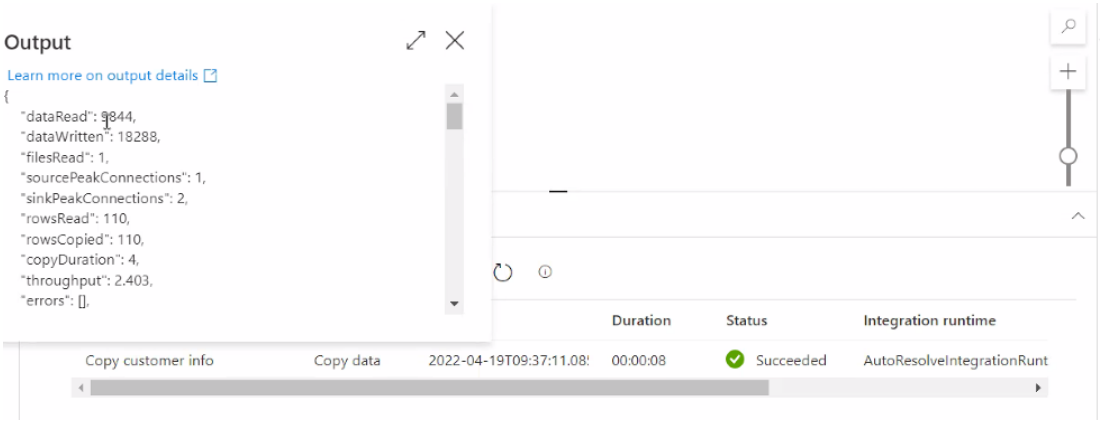

General
Source
Sink
Mapping
Settings
User properties
Sink dataset *
AzureSqlTable1
Open
New
Learn more
Write behavior
Insert
Upsert
Stored procedure
Bulk insert table lock
Yes
No
Table option
None
Auto create table
Pre-copy script
truncate table customerInfo
Write batch timeout
Write batch size
Max concurrent connections

Mapping:

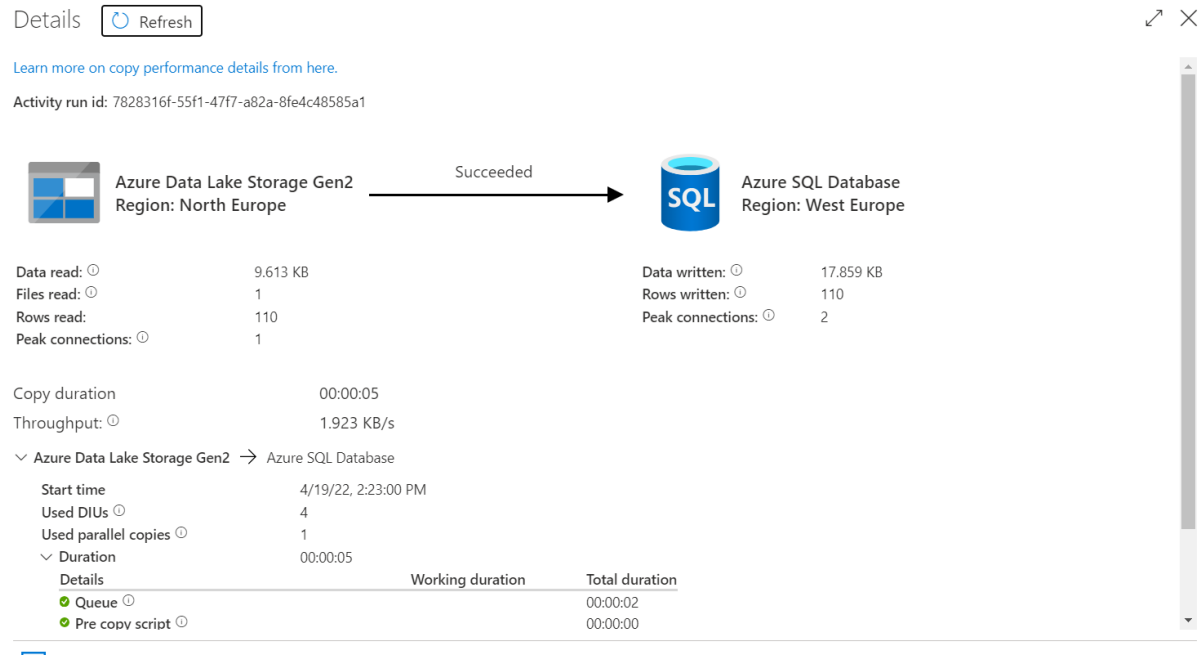

General
Source
Sink
Mapping
Settings
User properties
Type conversion settings
Import schemas
Preview source
New mapping
Clear
Reset
Delete
Source
Type
Destination
Type
UserName
String
username
varchar
Gender
String
gender
varchar
Phone
String
phone_number
varchar
Email
String
email
varchar
CreditCard
String
creditcard
varchar

Let’s debug the pipeline:

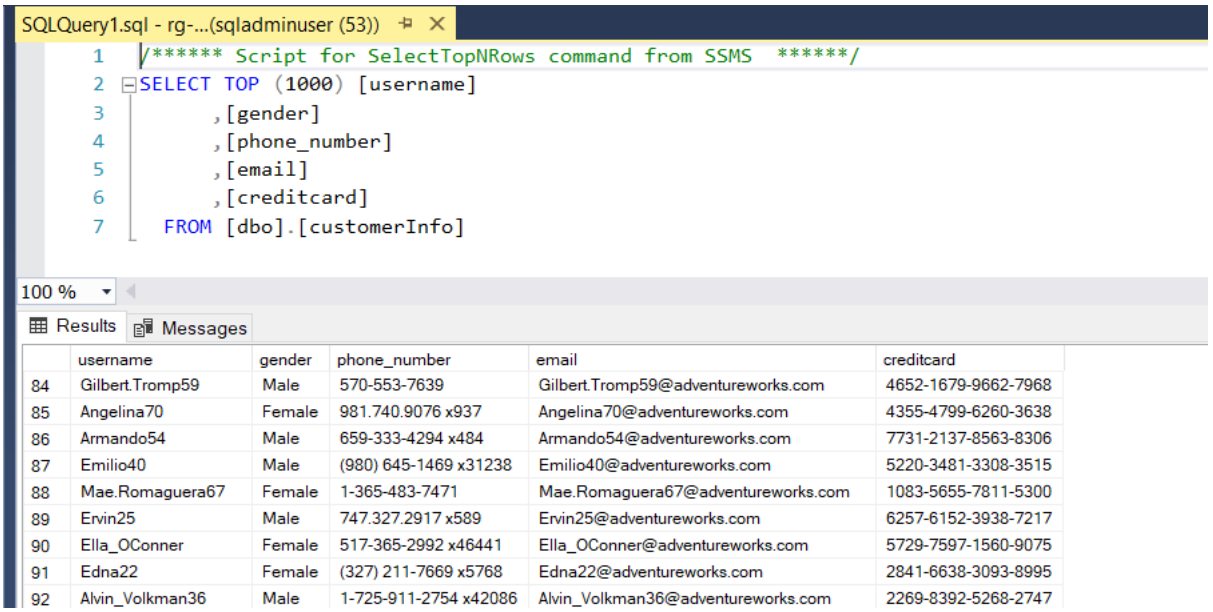




There are the details telling how data are passing from one location to another.

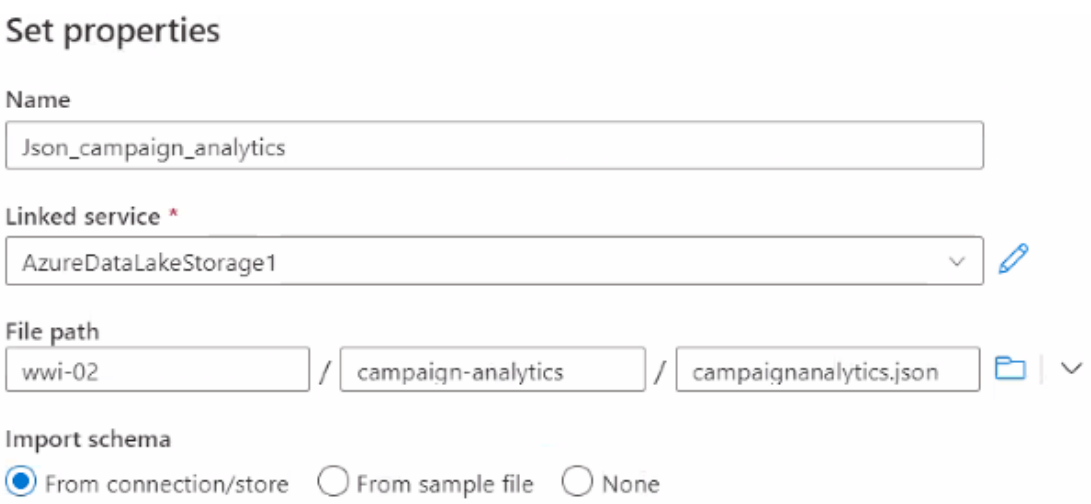


If we go to SQL server and do a select statement we will see the data are loaded there:



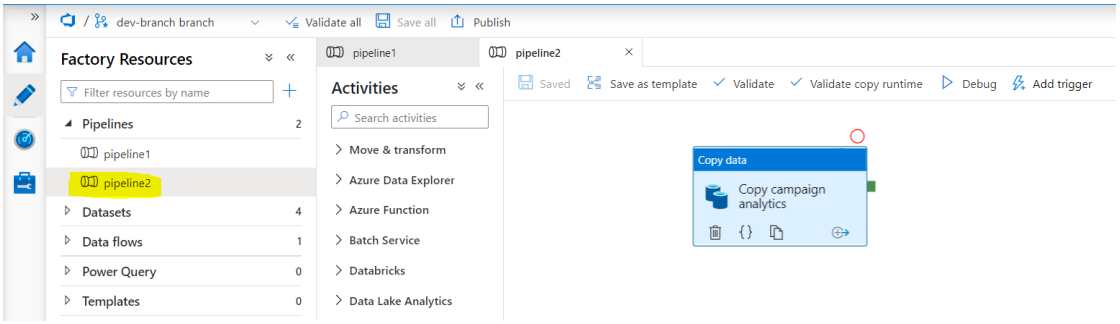
	username	gender	phone_number	email	creditcard
84	Gilbert.Tromp59	Male	570-553-7639	Gilbert.Tromp59@adventureworks.com	4652-1679-9662-7968
85	Angelina70	Female	981.740.9076 x937	Angelina70@adventureworks.com	4355-4799-6260-3638
86	Armando54	Male	659-333-4294 x484	Armando54@adventureworks.com	7731-2137-8563-8306
87	Emilio40	Male	(980) 645-1469 x31238	Emilio40@adventureworks.com	5220-3481-3308-3515
88	Mae.Romaguera67	Female	1-365-483-7471	Mae.Romaguera67@adventureworks.com	1083-5655-7811-5300
89	Ervin25	Male	747.327.2917 x589	Ervin25@adventureworks.com	6257-6152-3938-7217
90	Ella_OConner	Female	517-365-2992 x46441	Ella_OConner@adventureworks.com	5729-7597-1560-9075
91	Edna22	Female	(327) 211-7669 x5768	Edna22@adventureworks.com	2841-6638-3093-8995
92	Alvin_Volkman36	Male	1-725-911-2754 x42086	Alvin_Volkman36@adventureworks.com	2269-8392-5268-2747

Now let's import the json dataset.

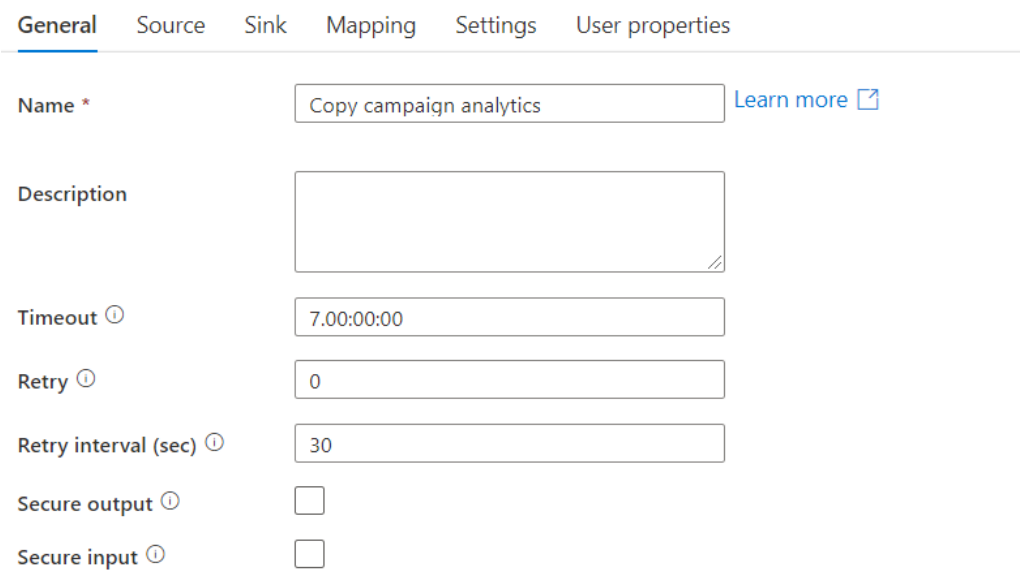


Create a new dataset to store the data:

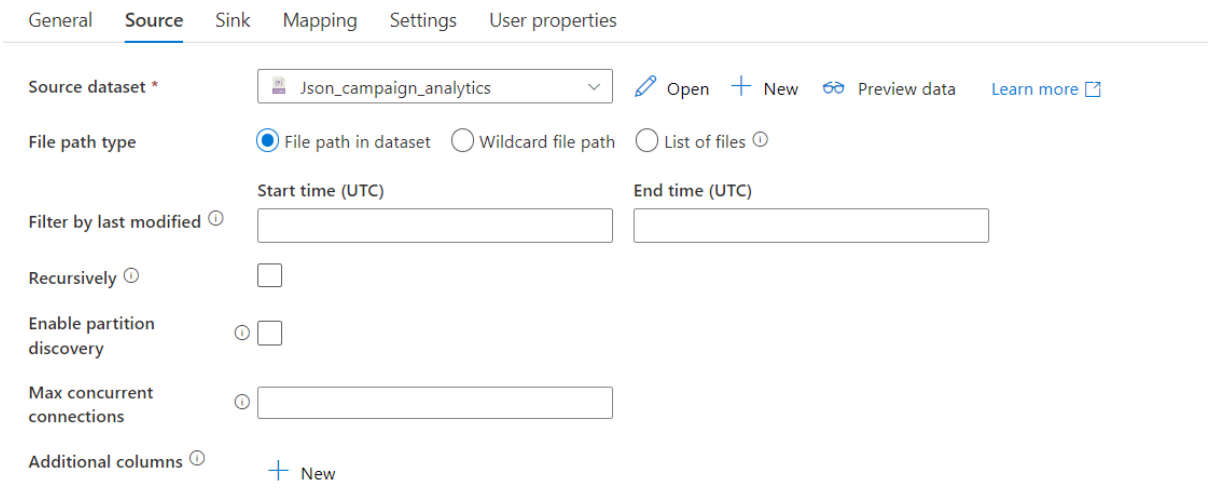
Here we create a new pipeline for loading the data to the SQL server.



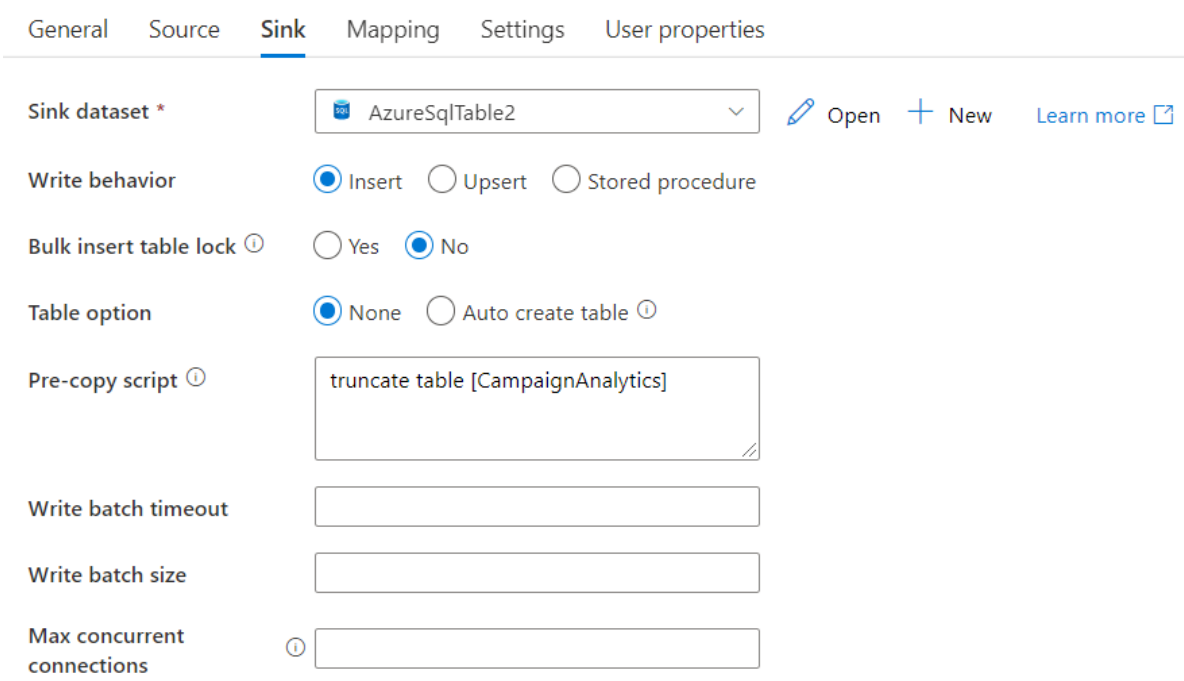
The configuration of the pipeline is as below:



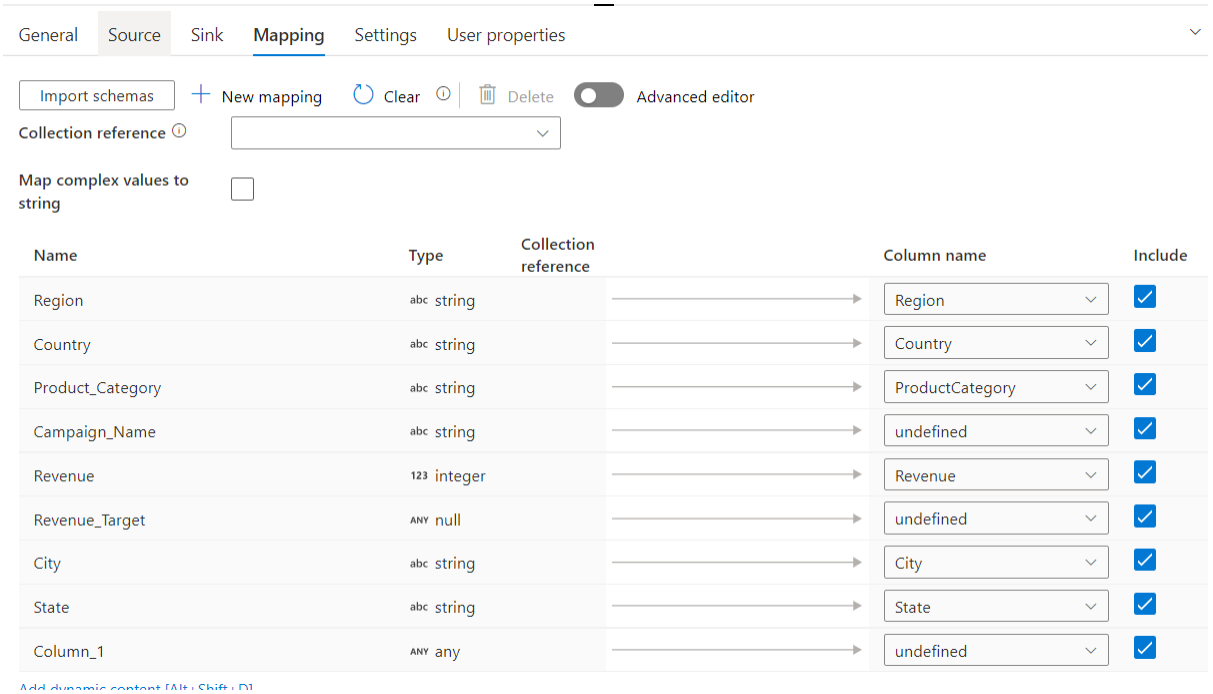
Source:



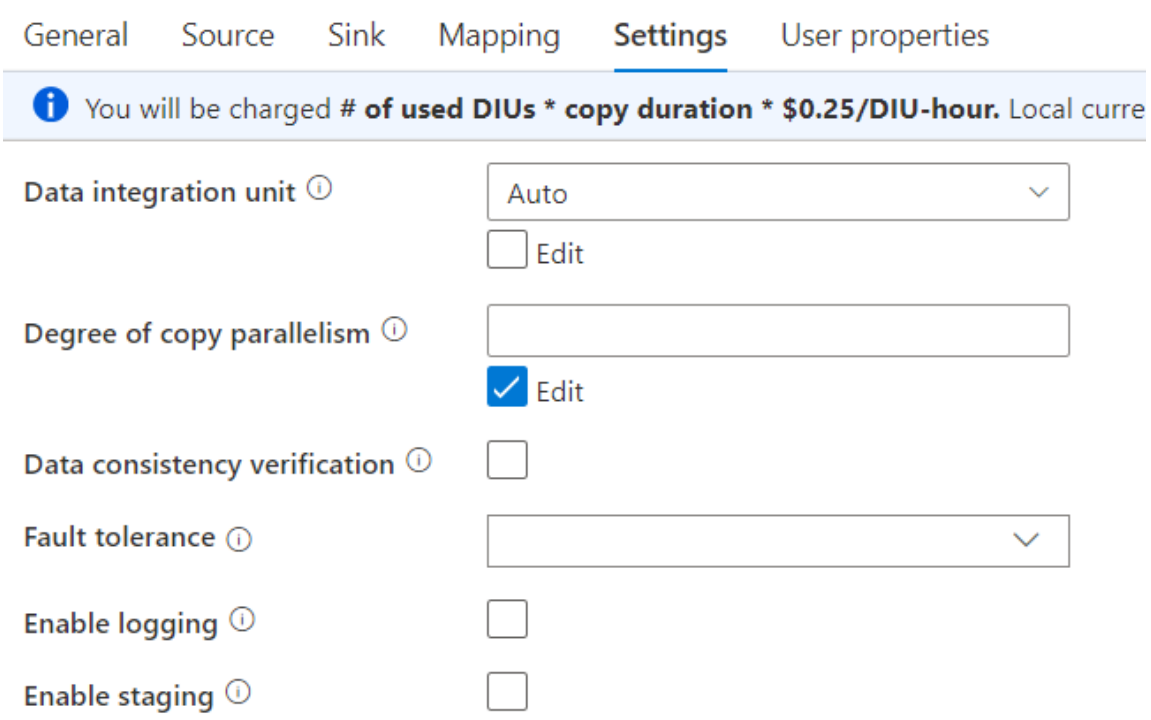
Sink:



Mapping:


General
Source
Sink
Mapping
Settings
User properties
Import schemas
New mapping
Clear
Delete
Advanced editor
Collection reference
Map complex values to string
Name
Type
Collection reference
Column name
Include
Region
abc string
Region
Country
abc string
Country
Product_Category
abc string
ProductCategory
Campaign_Name
abc string
undefined
Revenue
123 integer
Revenue
Revenue_Target
ANY null
undefined
City
abc string
City
State
abc string
State
Column_1
ANY any
undefined

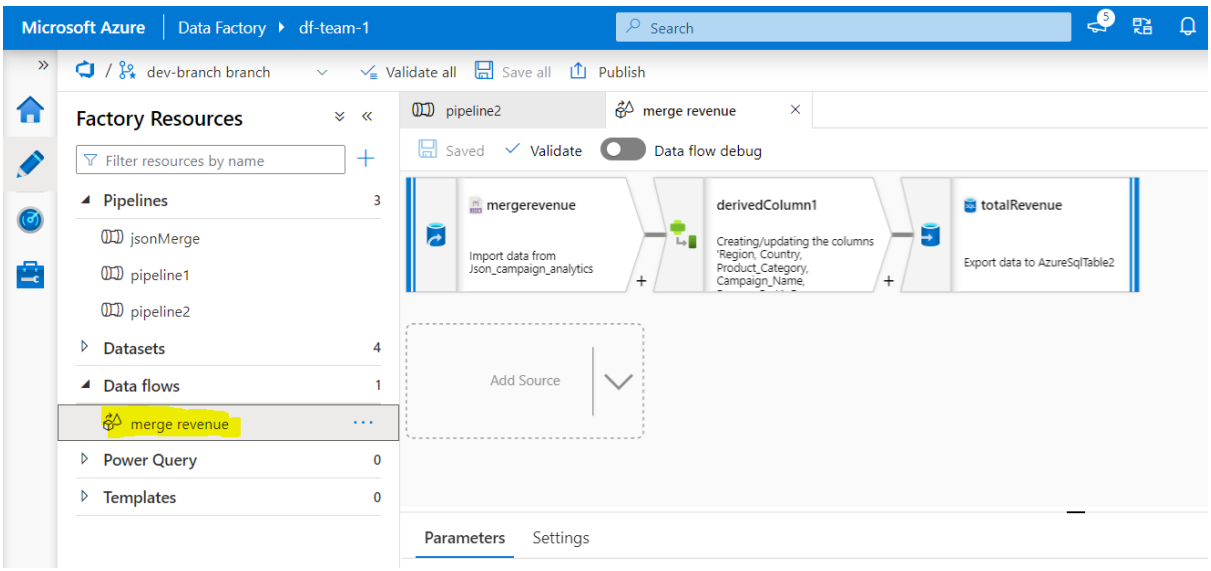
Setting:


General
Source
Sink
Mapping
Settings
User properties
You will be charged # of used DIUs * copy duration * $0.25/DIU-hour. Local curre
Data integration unit
Auto
Edit
Degree of copy parallelism
Edit
Data consistency verification
Fault tolerance
Enable logging
Enable staging

Now we will create a dataflow in azure in order to transform/clean the json file.

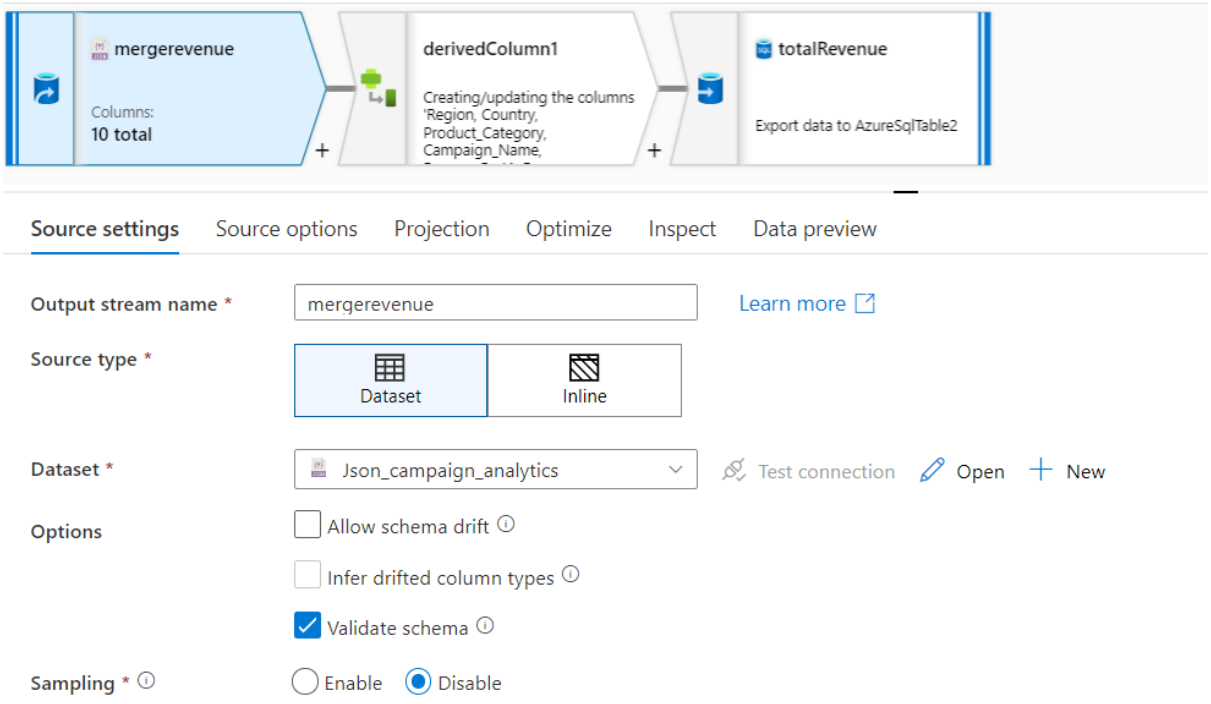
This is the original form of the json file, and as we can see there are some files which need to be transformed and merged.

For this we have created this data flow:

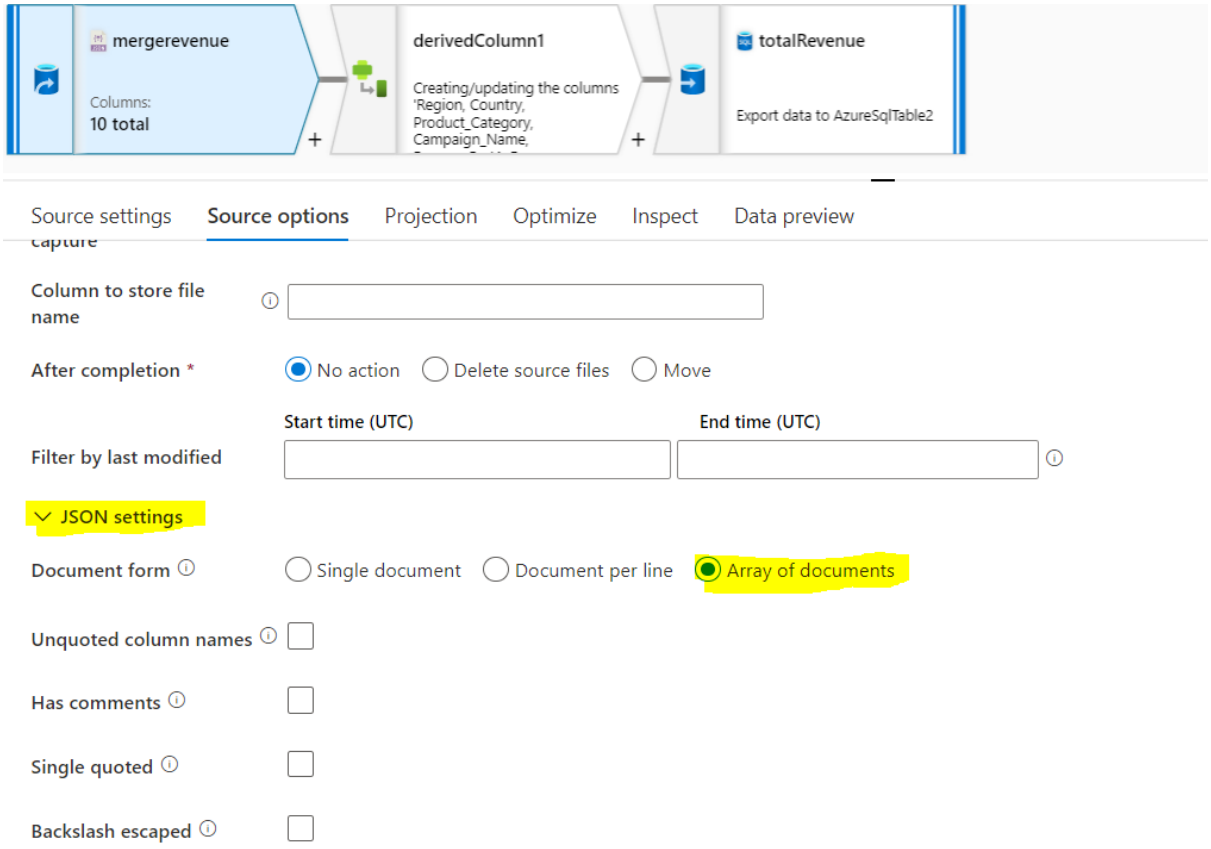


Merge revenue:

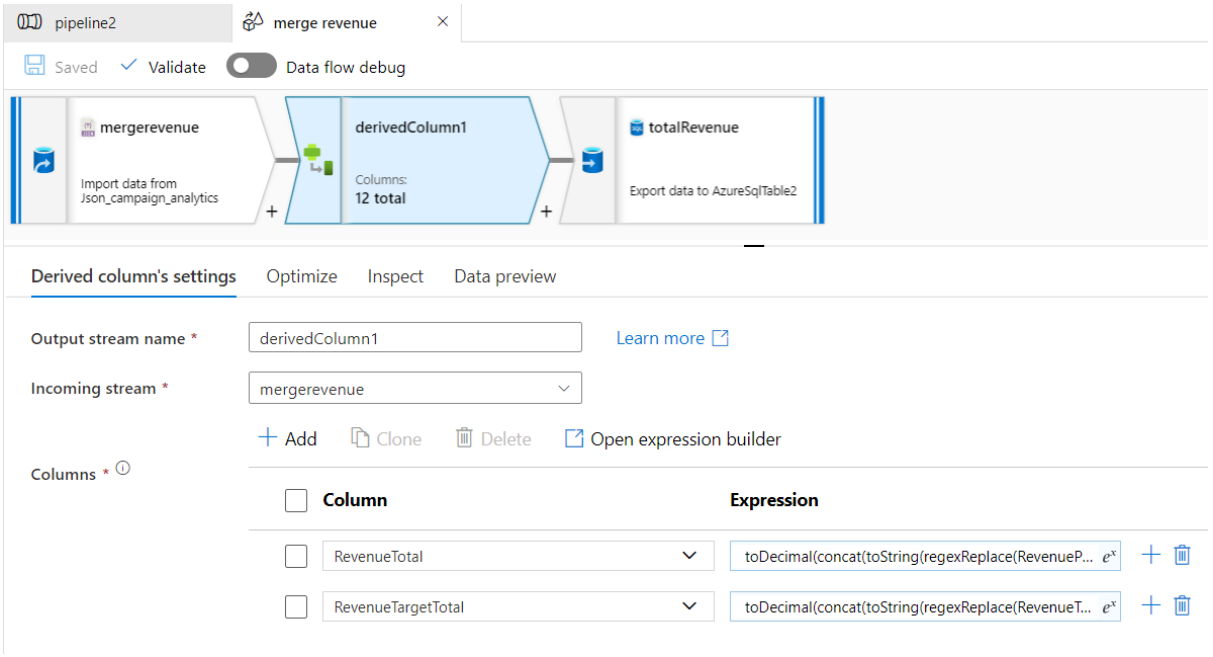
Source setting:



We should also do this configuration to read the the array of documents:

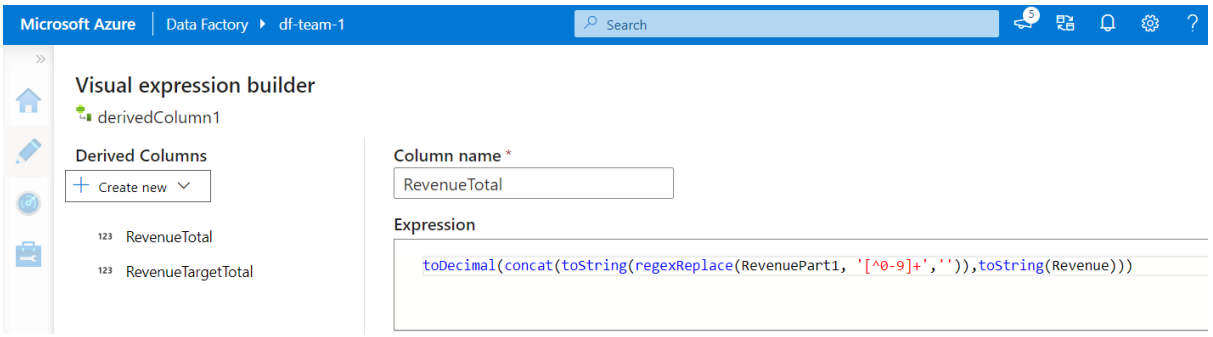


DerivedColumns configuration:



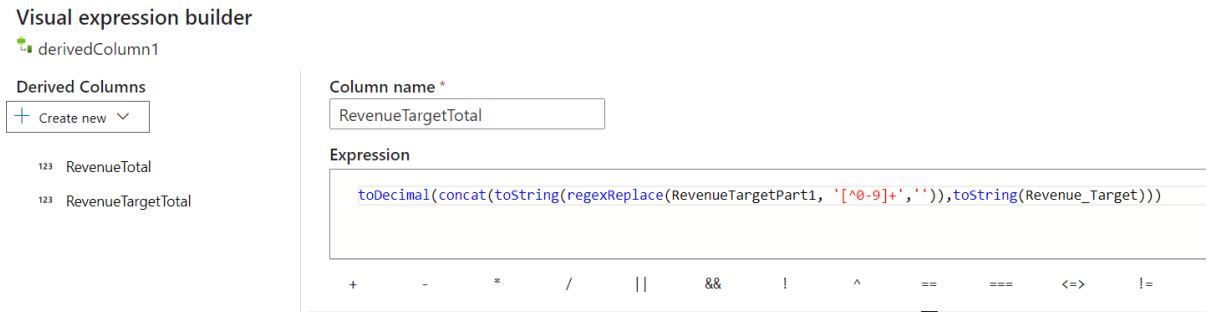
The function for RevenueTotal:

```
toDecimal(concat(toString(regexReplace(RevenuePart1,
'[^0-9]+',''),toString(Revenue)))
```

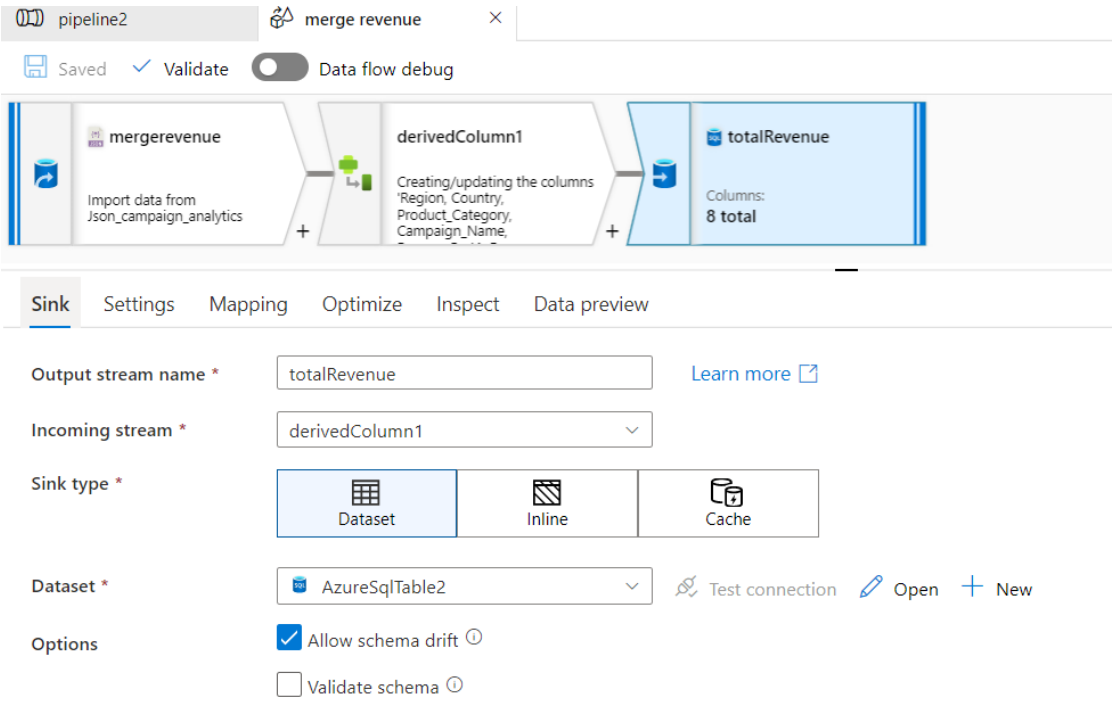


The function for RevenueTargetTotal:

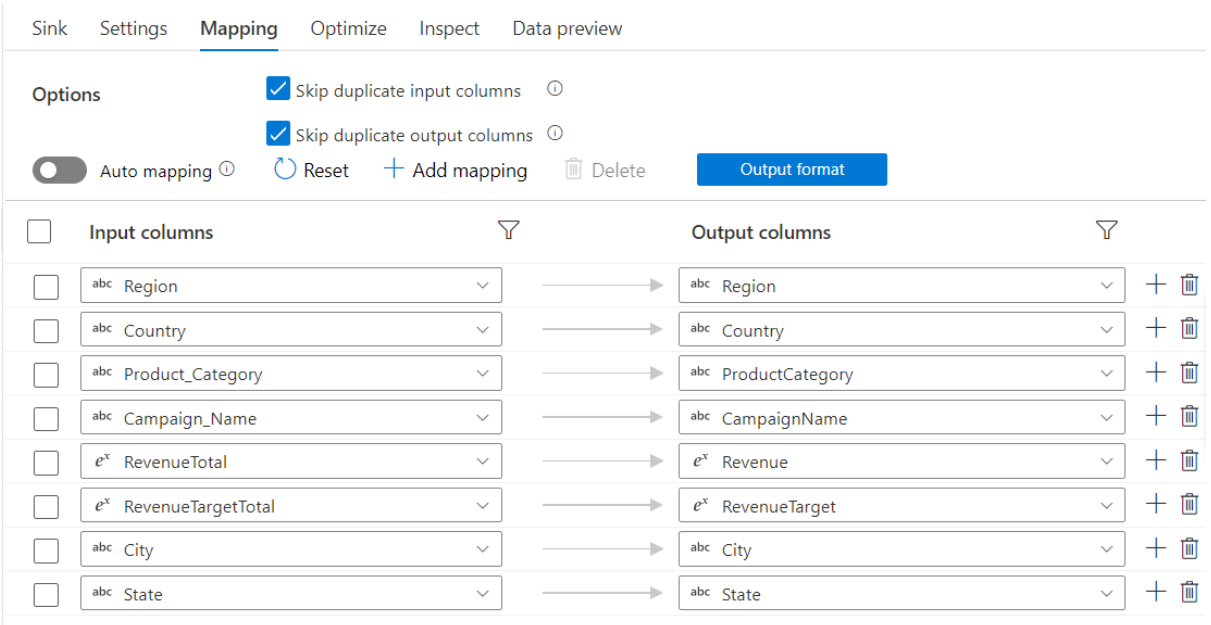
```
toDecimal(concat(toString(regexReplace(RevenueTargetPart1,
'[^0-9]+',''),toString(Revenue_Target)))
```



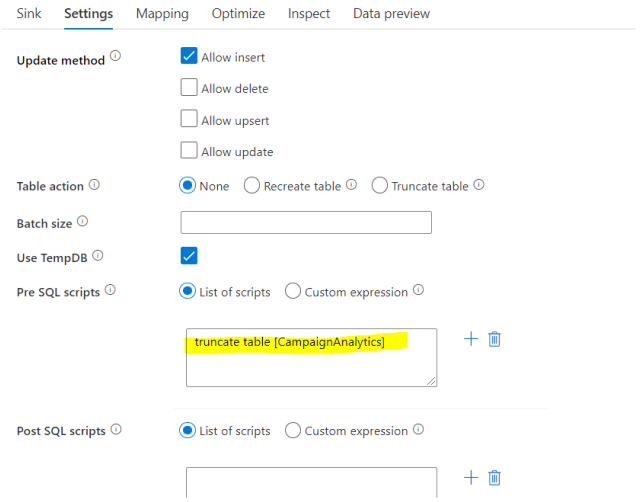
TotalRevenue:



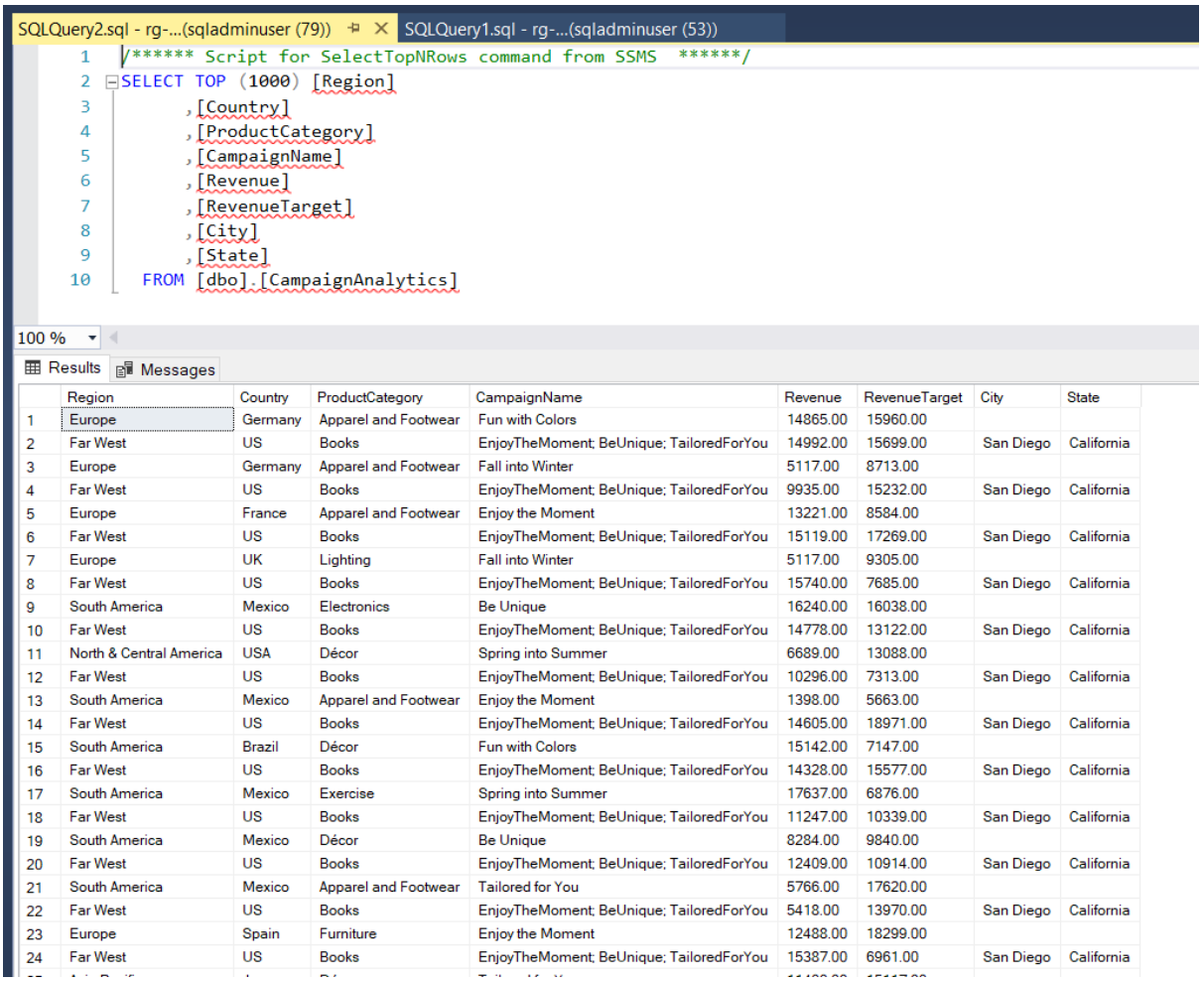
Mapping: Here we removed some columns and added the derived columns.



In settings we have to add this:



In SQL server we will see the data properly loaded:

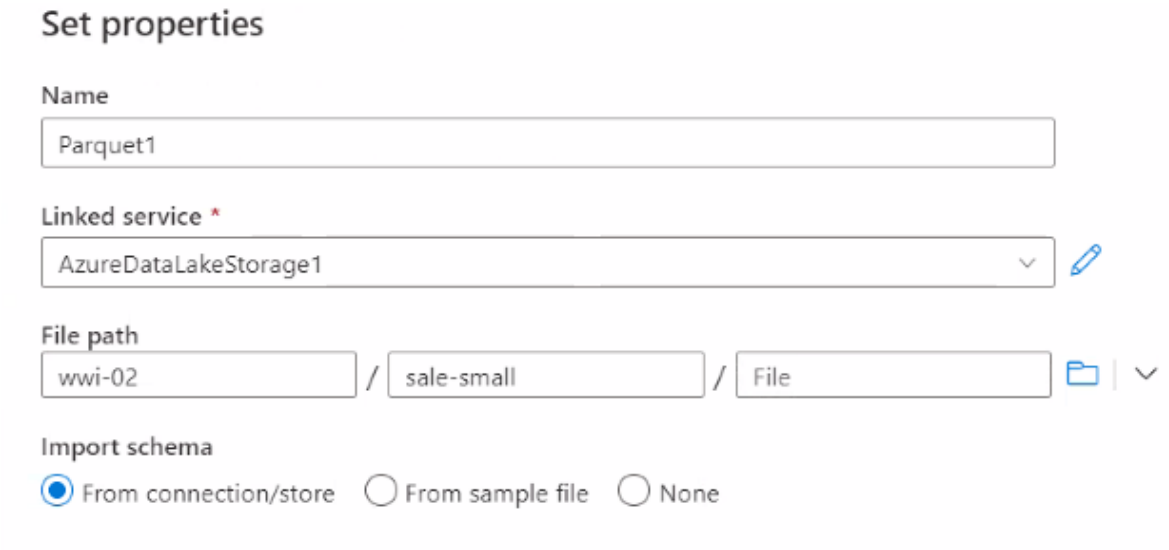


	Region	Country	ProductCategory	CampaignName	Revenue	RevenueTarget	City	State
1	Europe	Germany	Apparel and Footwear	Fun with Colors	14865.00	15960.00		
2	Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	14992.00	15699.00	San Diego	California
3	Europe	Germany	Apparel and Footwear	Fall into Winter	5117.00	8713.00		
4	Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	9935.00	15232.00	San Diego	California
5	Europe	France	Apparel and Footwear	Enjoy the Moment	13221.00	8584.00		
6	Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	15119.00	17269.00	San Diego	California
7	Europe	UK	Lighting	Fall into Winter	5117.00	9305.00		
8	Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	15740.00	7685.00	San Diego	California
9	South America	Mexico	Electronics	Be Unique	16240.00	16038.00		
10	Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	14778.00	13122.00	San Diego	California
11	North & Central America	USA	Décor	Spring into Summer	6689.00	13088.00		
12	Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	10296.00	7313.00	San Diego	California
13	South America	Mexico	Apparel and Footwear	Enjoy the Moment	1398.00	5663.00		
14	Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	14605.00	18971.00	San Diego	California
15	South America	Brazil	Décor	Fun with Colors	15142.00	7147.00		
16	Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	14328.00	15577.00	San Diego	California
17	South America	Mexico	Exercise	Spring into Summer	17637.00	6876.00		
18	Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	11247.00	10339.00	San Diego	California
19	South America	Mexico	Décor	Be Unique	8284.00	9840.00		
20	Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	12409.00	10914.00	San Diego	California
21	South America	Mexico	Apparel and Footwear	Tailored for You	5766.00	17620.00		
22	Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	5418.00	13970.00	San Diego	California
23	Europe	Spain	Furniture	Enjoy the Moment	12488.00	18299.00		
24	Far West	US	Books	EnjoyTheMoment; BeUnique; TailoredForYou	15387.00	6961.00	San Diego	California

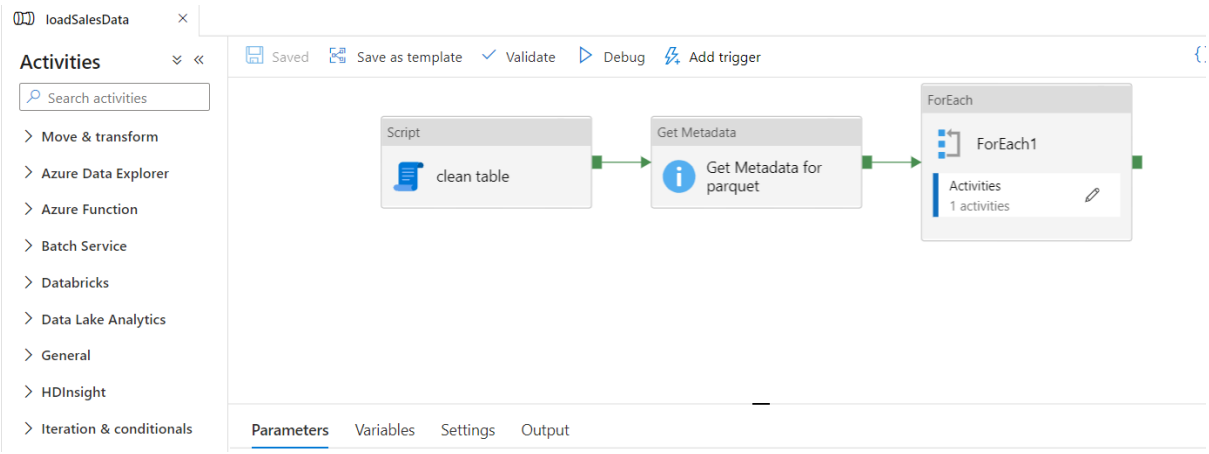
Create a table in SQL server for the parque file:

..
…
..

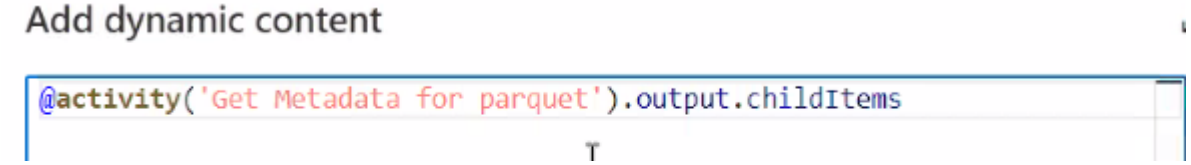
In Azure create a linked service to connect to the files we want to load.



Now let's create a pipeline for loading the parque file:

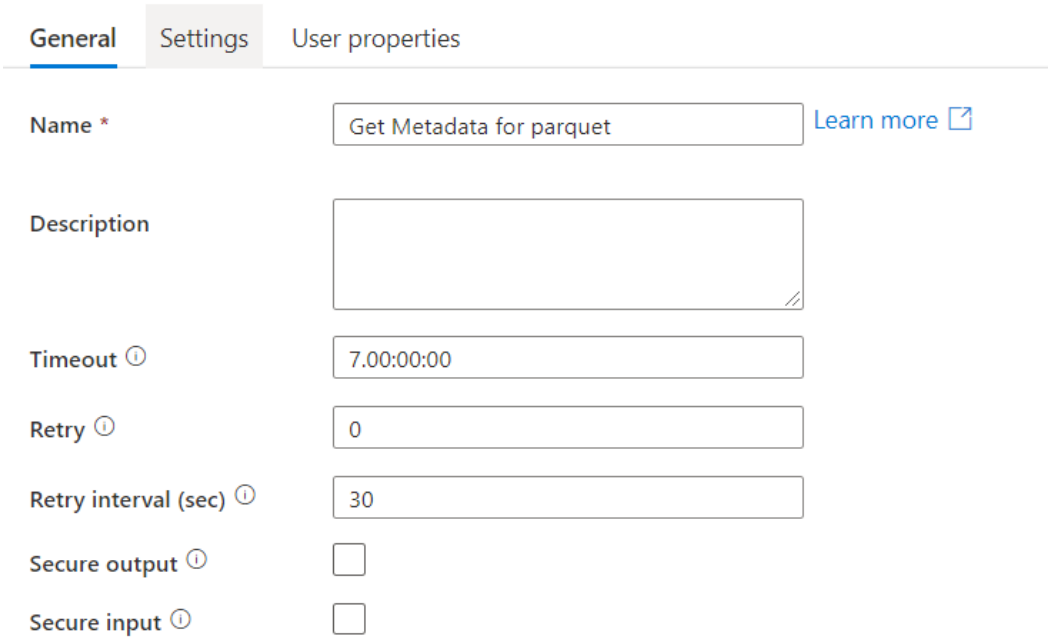


For the Items add this dynamic activity:

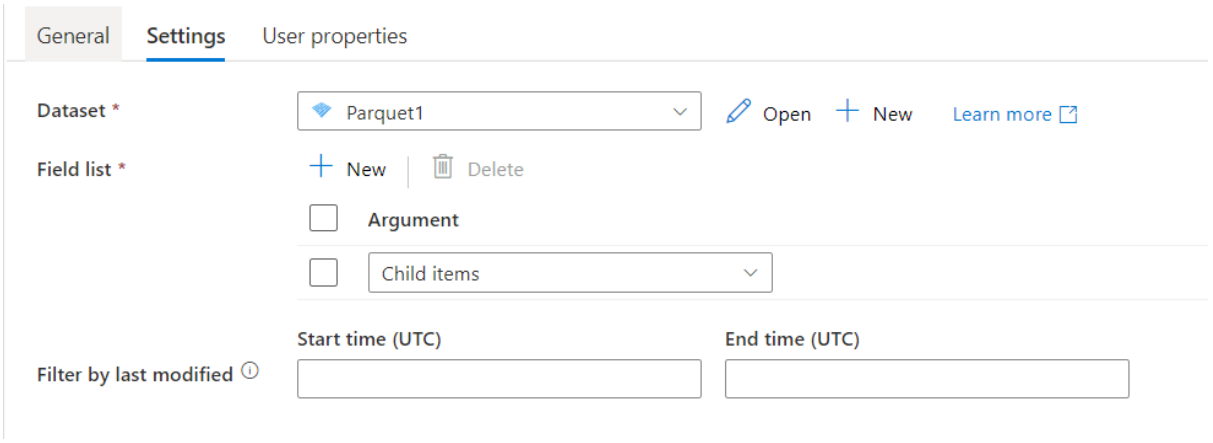


The configuration for the activities in the pipeline for parquet files is as below.

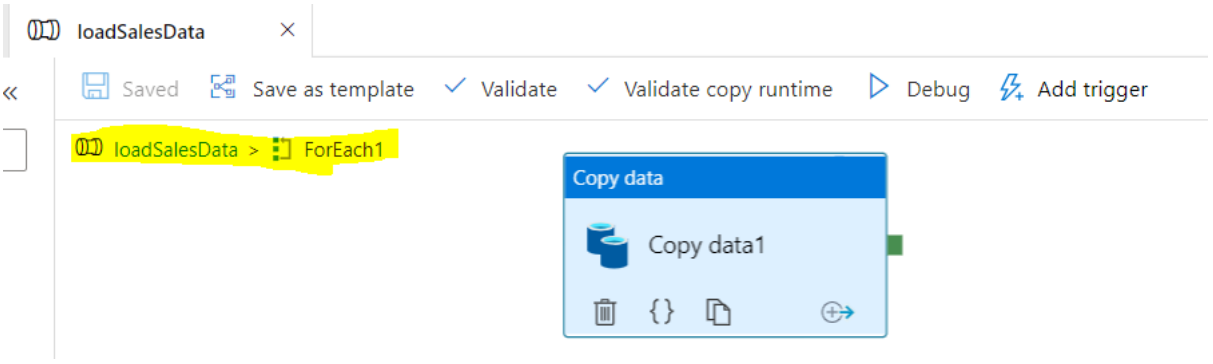
General:



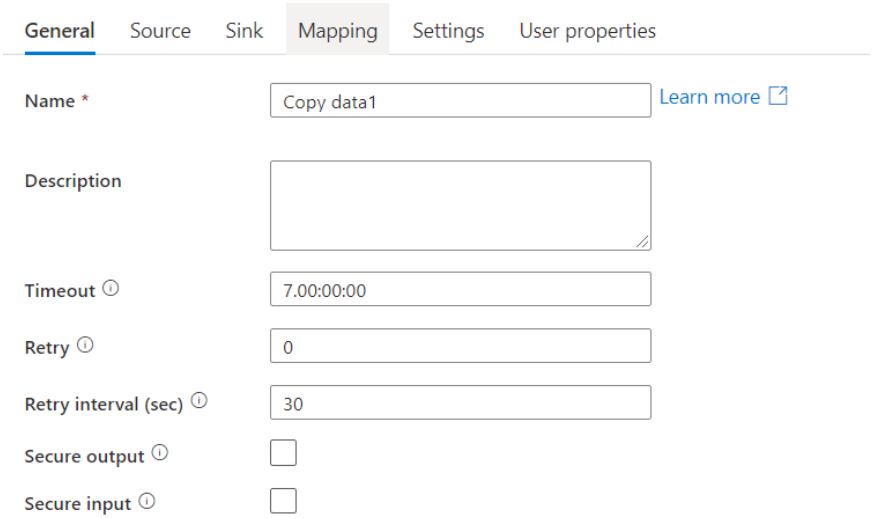
Setting:



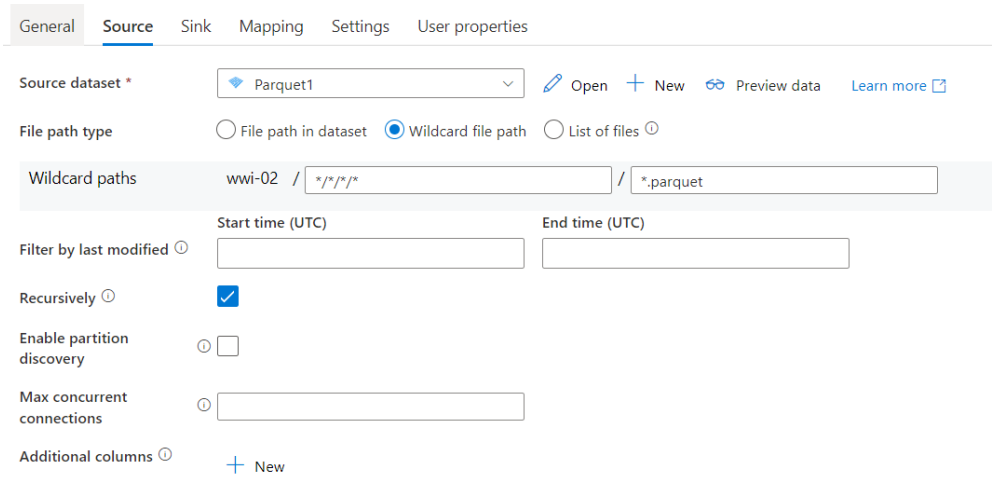
For Each loop configuration:



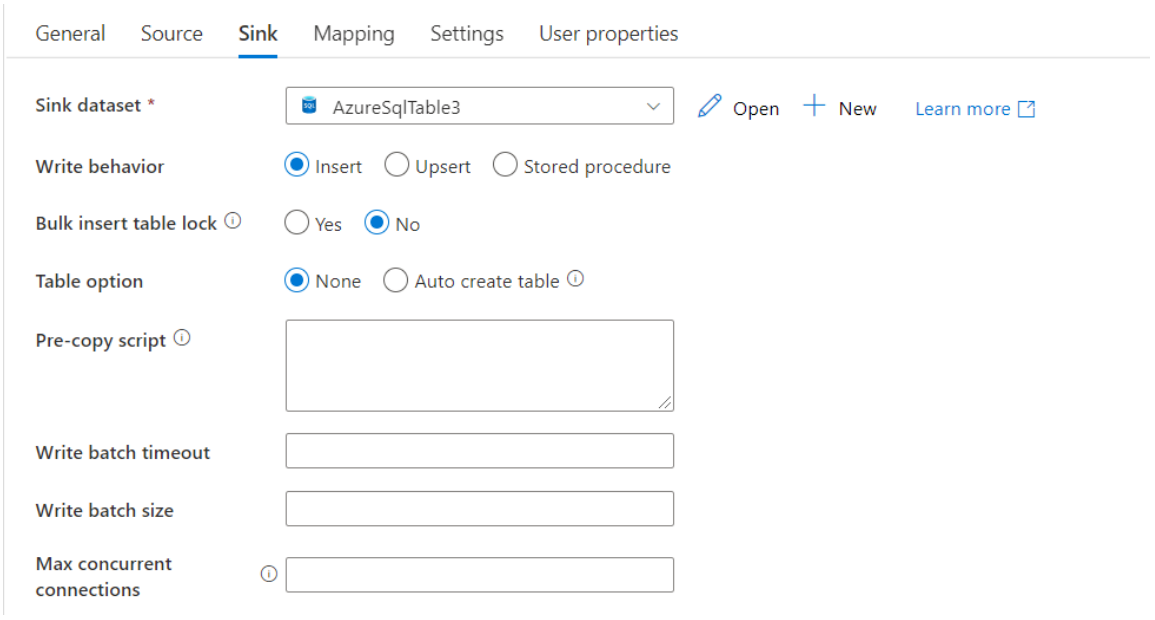
General:



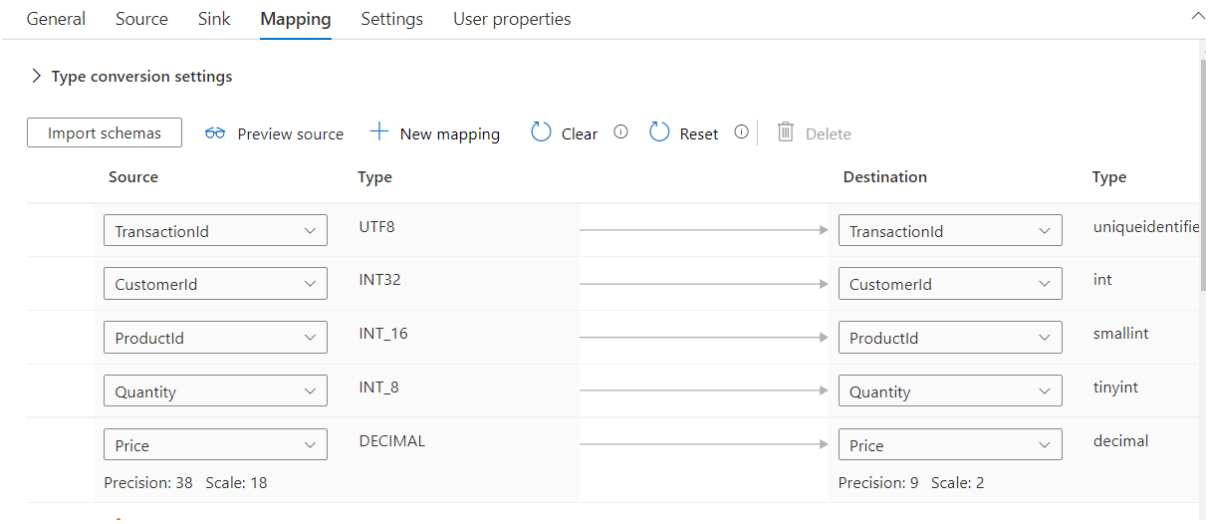
Source:



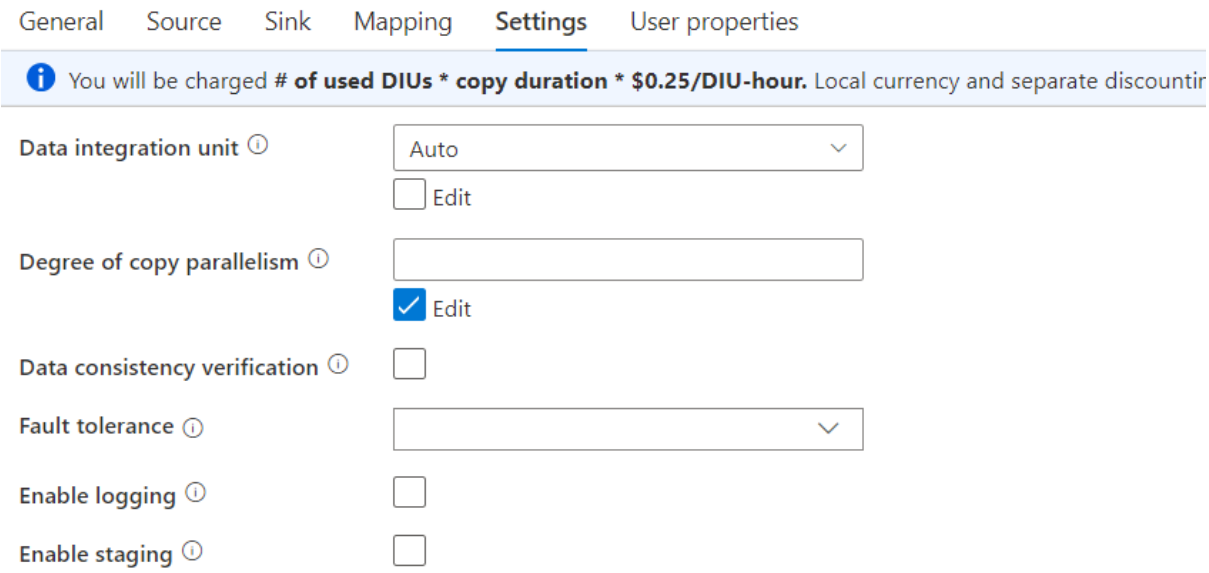
Sink:



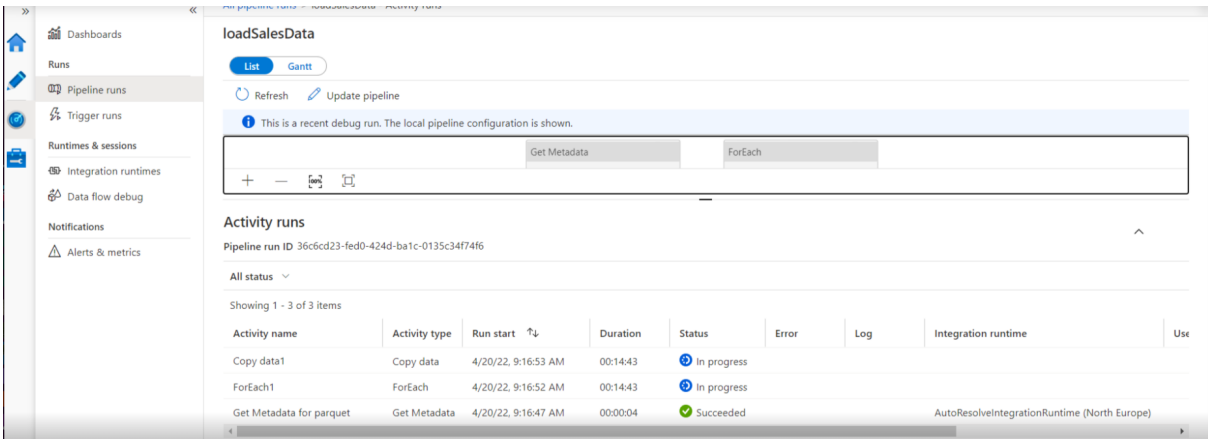
Mapping:



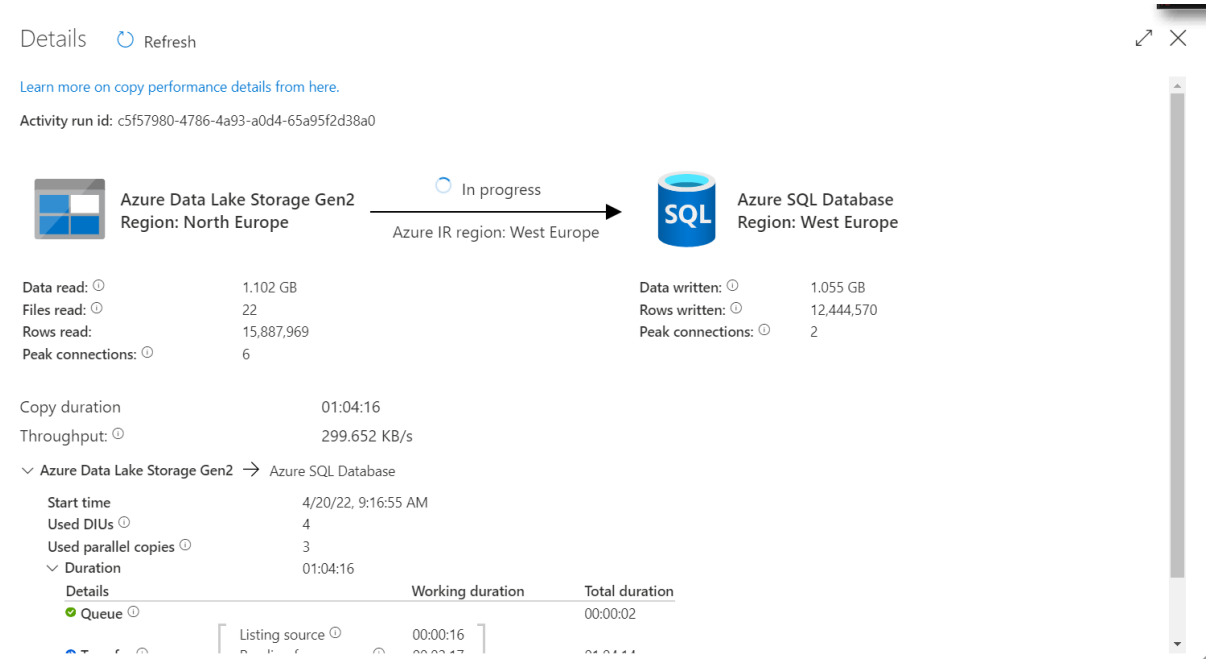
Settings:



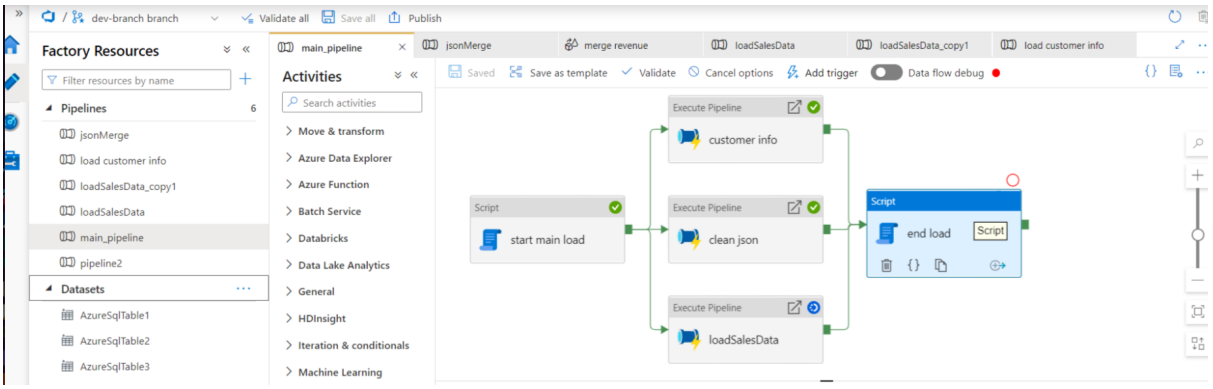
Now we will run and debug the pipeline.



The execution of the pipeline for loading parquet file takes time and the progress till now is as below:



We have also added a main pipeline that starts all the pipelines which will run in parallel and then it records in a table the starting and the ending time for the tasks.



Bonus task:

A. (Bonus points) Connect Streamlit to data source (3pt)

B. (Bonus points) Present Dashboard with the data (3pt)

For this we will need to install the Streamlit library. For installing this we use the command:

```
pip install streamlit
```

We might also need these two libraries:

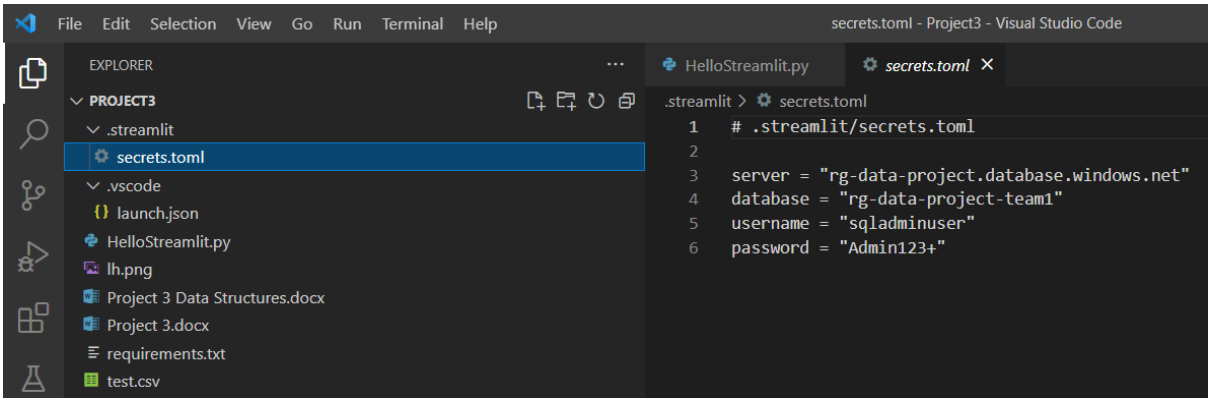
```
pandas
```

```
plotly
```

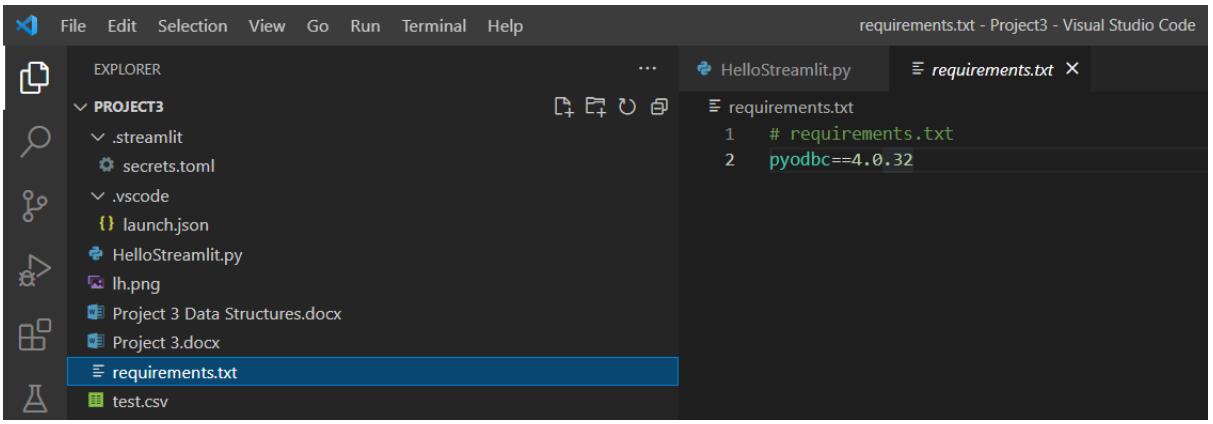
For this project we have decided to visualise the data related to **Campaign Analytics.**

For connecting streamlit to SQL Server we have done the following:

1. Added the credentials to the local apps secret.Our local Streamlit app will read secrets from a file `.streamlit/secrets.toml` in our app's root directory.



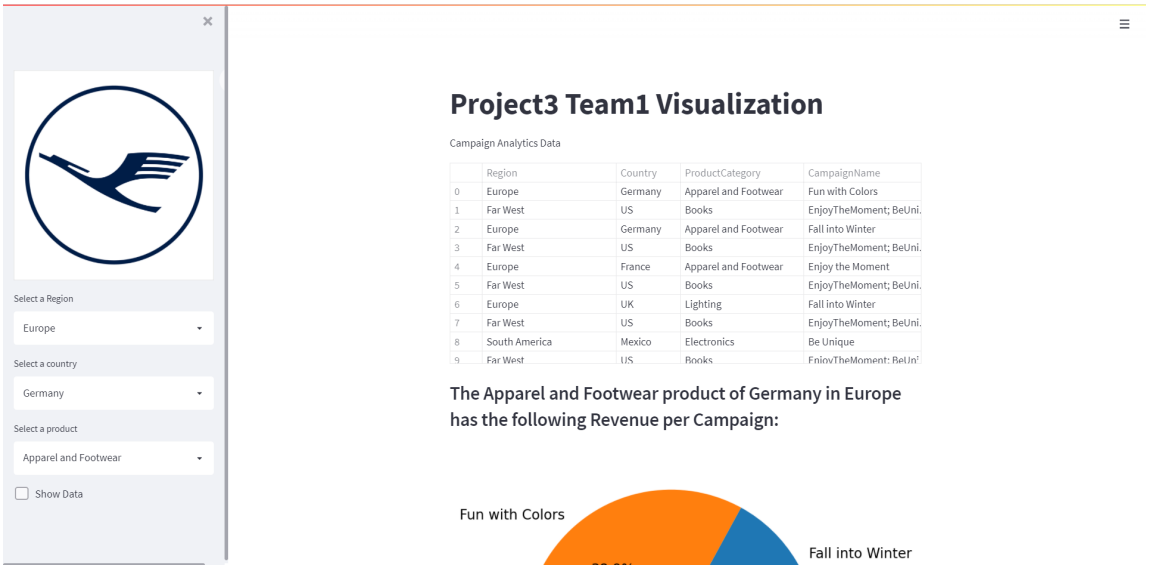
2. Added pyodbc to our requirements file



3. Modified our code below to establish the connection.

```
def init_connection():

    return pyodbc.connect("DRIVER={ODBC Driver 17 for SQL
Server};SERVER=" + st.secrets["server"] + ";DATABASE=" +
st.secrets["database"] + ";UID="

        + st.secrets["username"]

        + ";PWD="

        + st.secrets["password"]

    )



conn = init_connection()

chunk_test = pd.read_sql('SELECT * from CampaignAnalytics', conn)
```

The output:



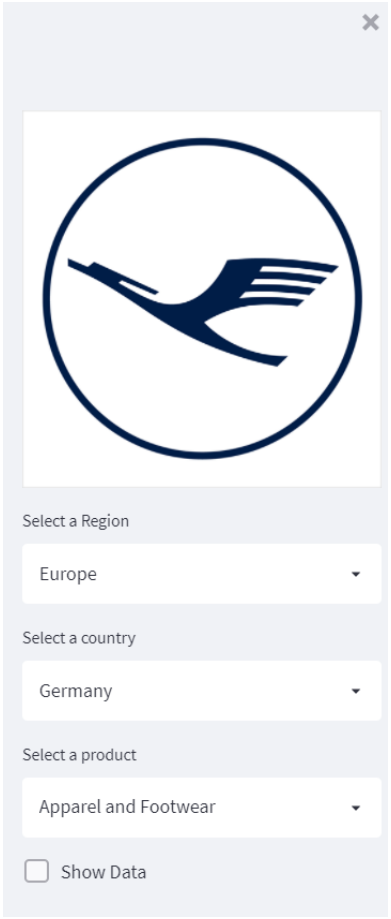
In the beginning of the page, we have displayed the dataset, which we transformed into Azure Data Factory.

Project3 Team1 Visualization

Campaign Analytics Data

	Region	Country	ProductCategory	CampaignName
20	South America	Mexico	Apparel and Footwear	Tailored for You
21	Far West	US	Books	EnjoyTheMoment; BeUni.
22	Europe	Spain	Furniture	Enjoy the Moment
23	Far West	US	Books	EnjoyTheMoment; BeUni.
24	Asia Pacific	Japan	Décor	Tailored for You
25	Far West	US	Books	EnjoyTheMoment; BeUni.
26	Europe	Germany	Electronics	Fall into Winter
27	Far West	US	Books	EnjoyTheMoment; BeUni.
28	Europe	Italy	Lighting	Get Sporty
29	Far West	US	Books	EnjoyTheMoment; BeUn

Below is given the side menu, which helps you to filter the data according to your preferences. You are able to select the region, country and the product.

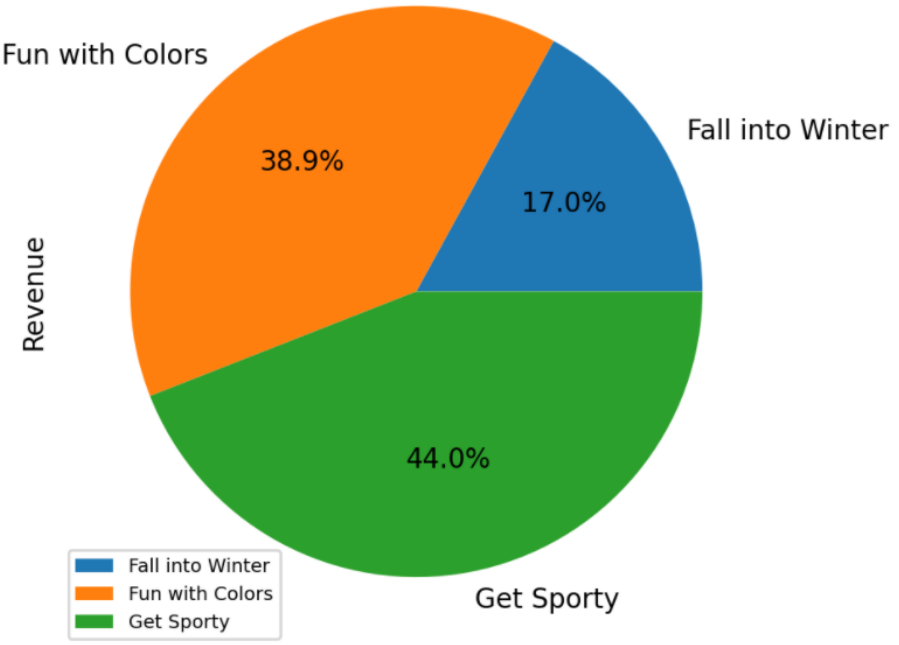


Clicking on the **Show Data** option, the data will be displayed in a tabular format for the selected filters.

The Apparel and Footwear product of Germany in Europe has the following Revenue per Campaign:

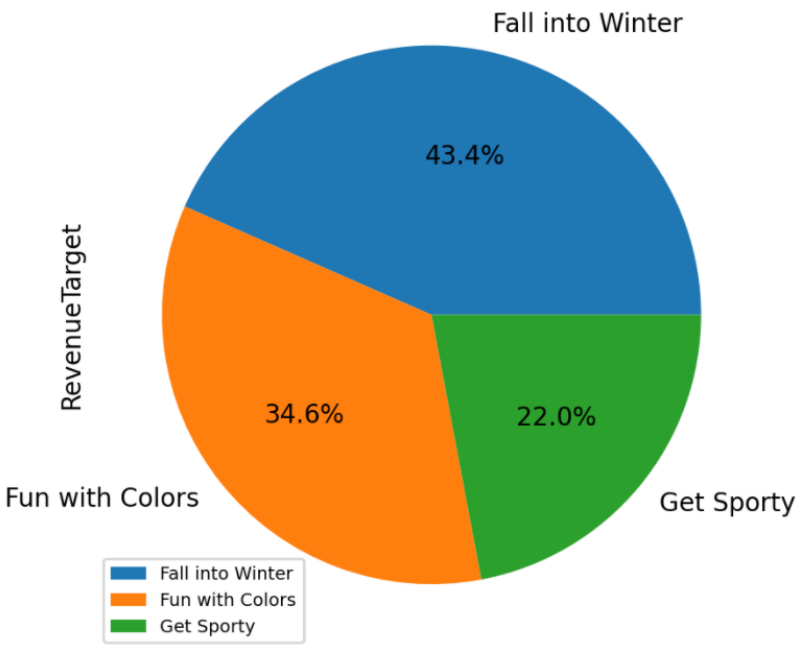
	Region	Country	ProductCategory	CampaignName	Revenue	Reve
0	Europe	Germany	Apparel and Footwear	Fun with Colors	14,865.0000	15
2	Europe	Germany	Apparel and Footwear	Fall into Winter	5,117.0000	8
38	Europe	Germany	Apparel and Footwear	Get Sporty	16,810.0000	10
57	Europe	Germany	Apparel and Footwear	Fall into Winter	1,383.0000	11
125	Europe	Germany	Apparel and Footwear	Fun with Colors	14,865.0000	15
127	Europe	Germany	Apparel and Footwear	Fall into Winter	5,117.0000	8
163	Europe	Germany	Apparel and Footwear	Get Sporty	16,810.0000	10
180	Europe	Germany	Apparel and Footwear	Fall into Winter	1,383.0000	11

Below we will see the data displayed with a pie chart.

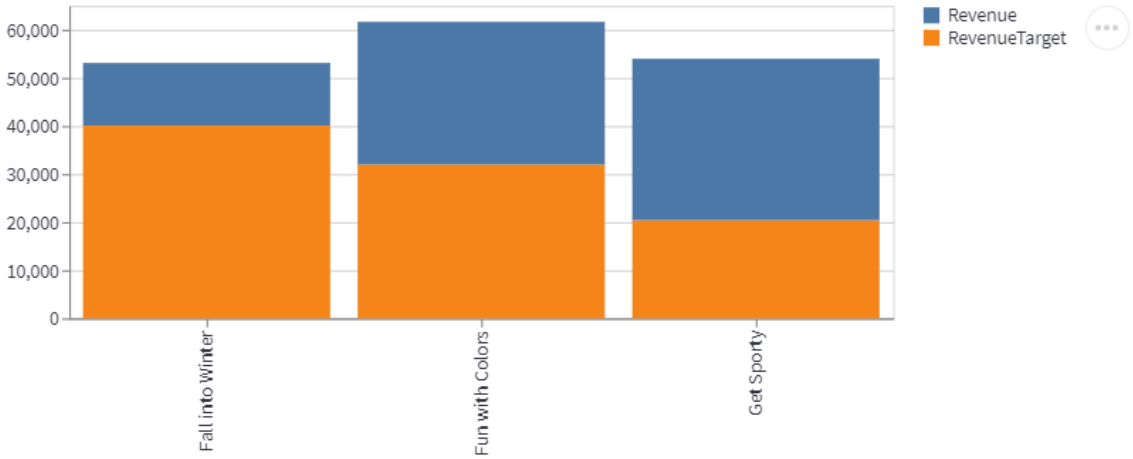


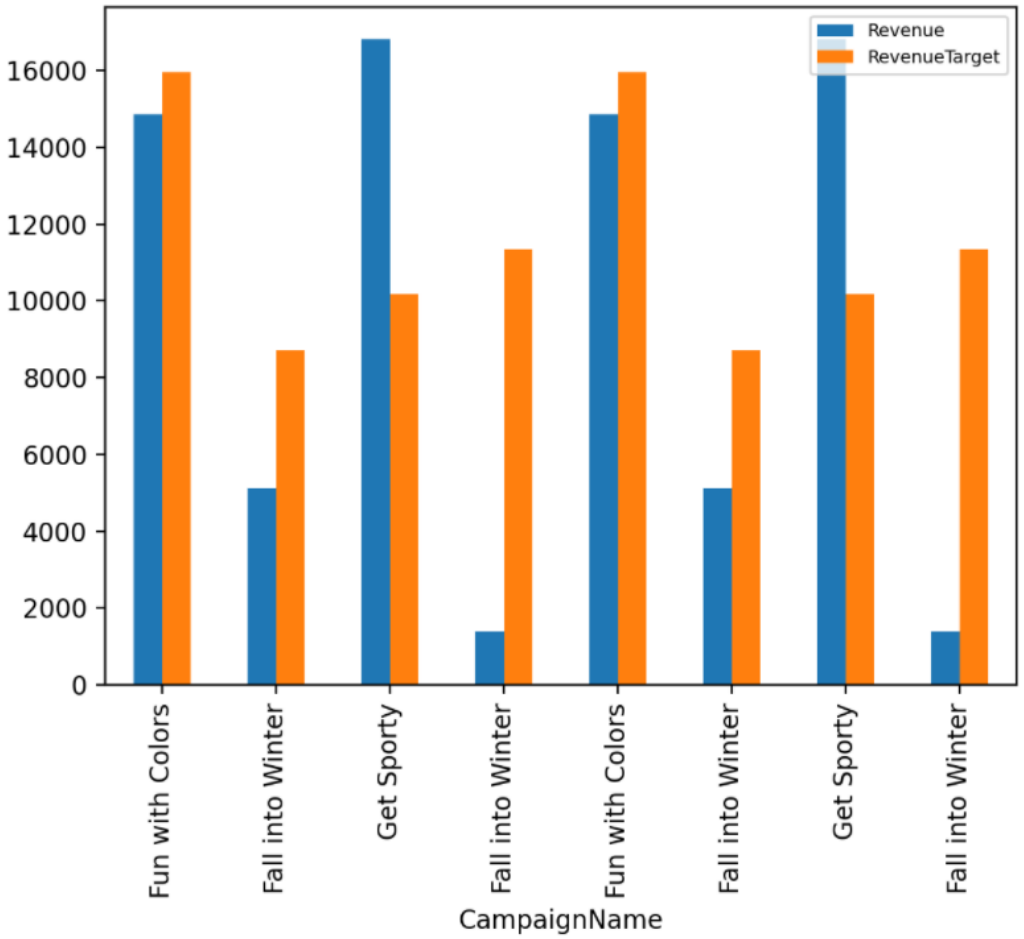
Below is displayed the Revenue Target per campaign based on the filters.

The Apparel and Footwear product of Germany in Europe has the following RevenueTarget per Campaign:

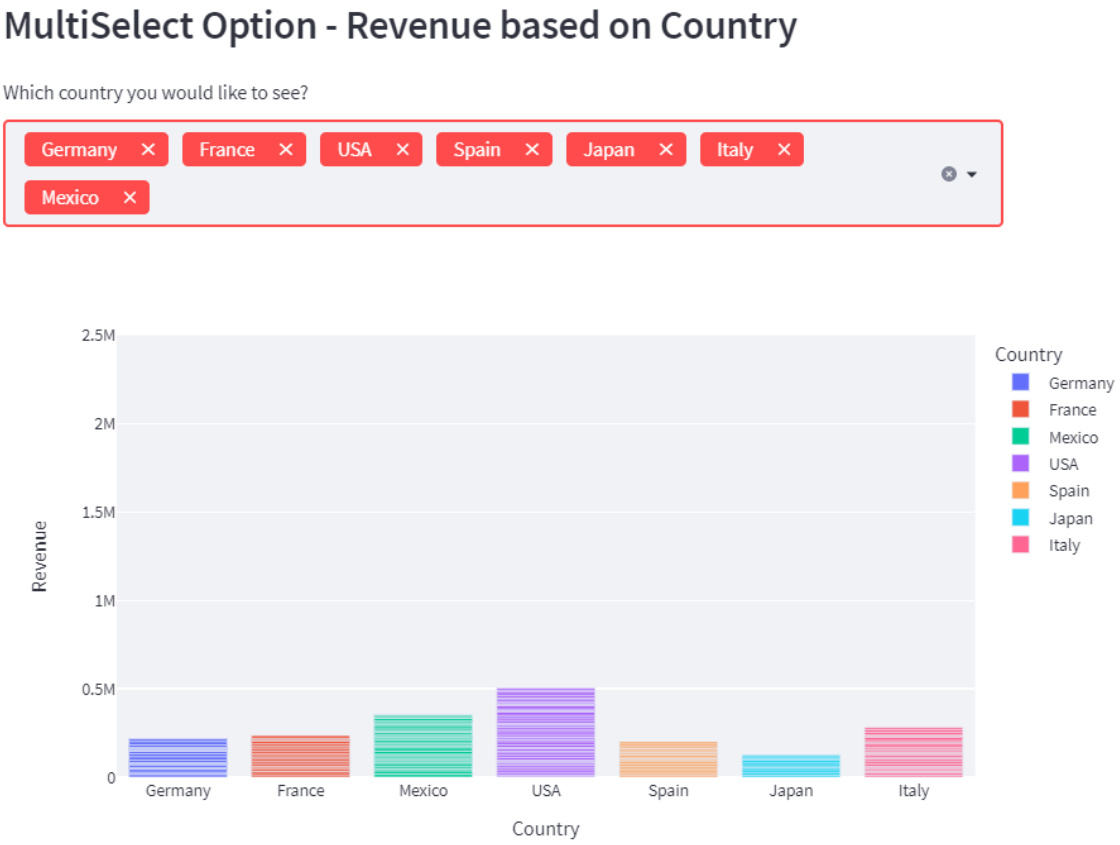


Another way we used to display to the data:

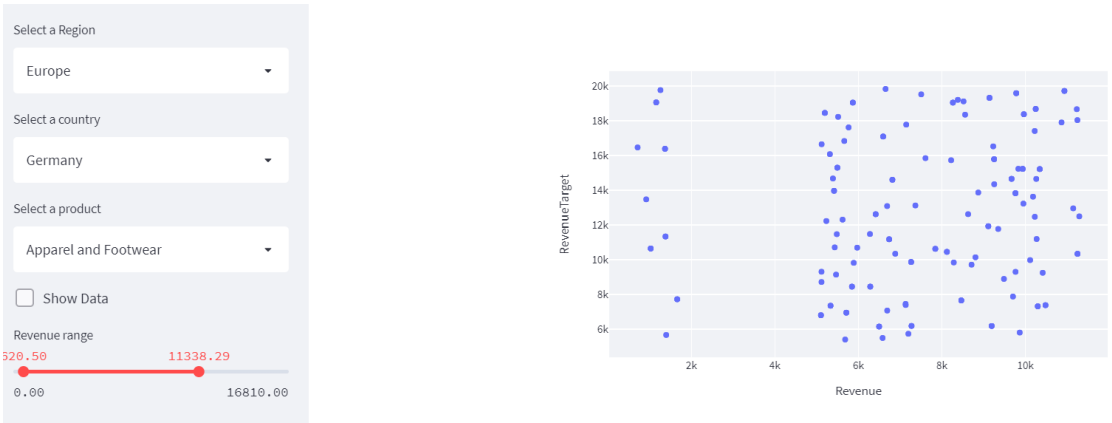




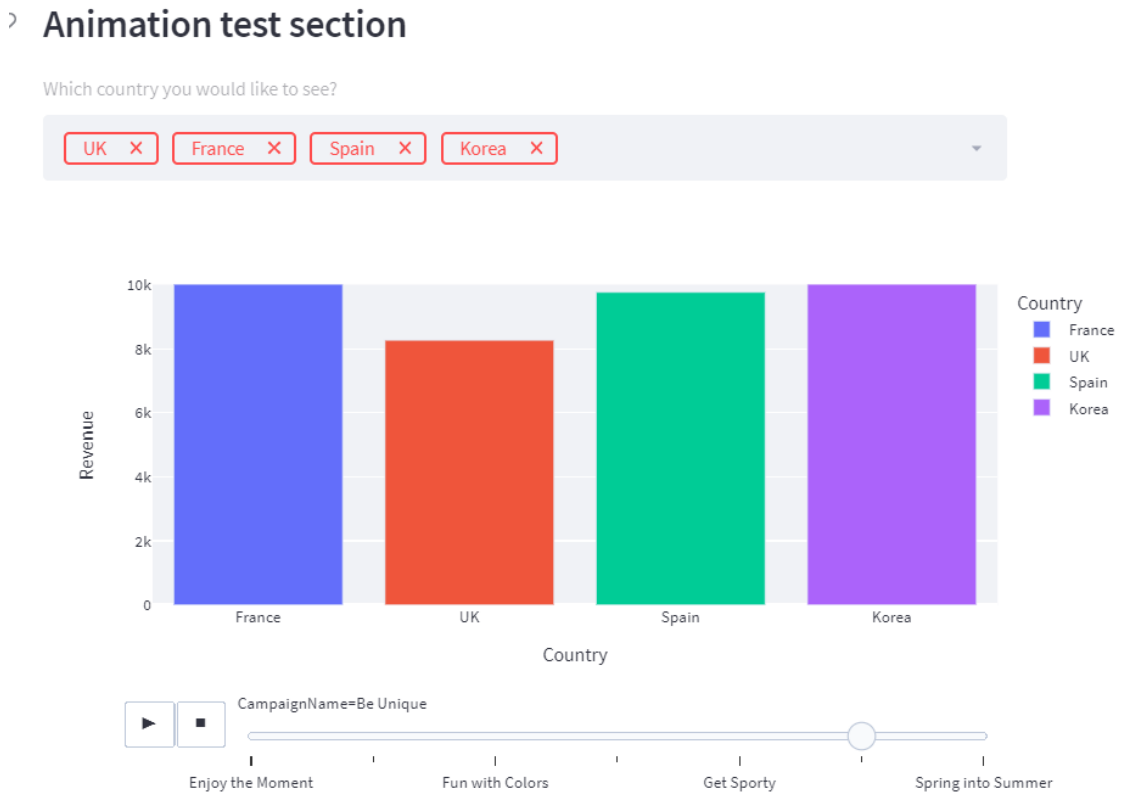
We have displayed the revenue based on the country. This is a multiselect option which allows us to select many countries and display the revenue for the selected countries.



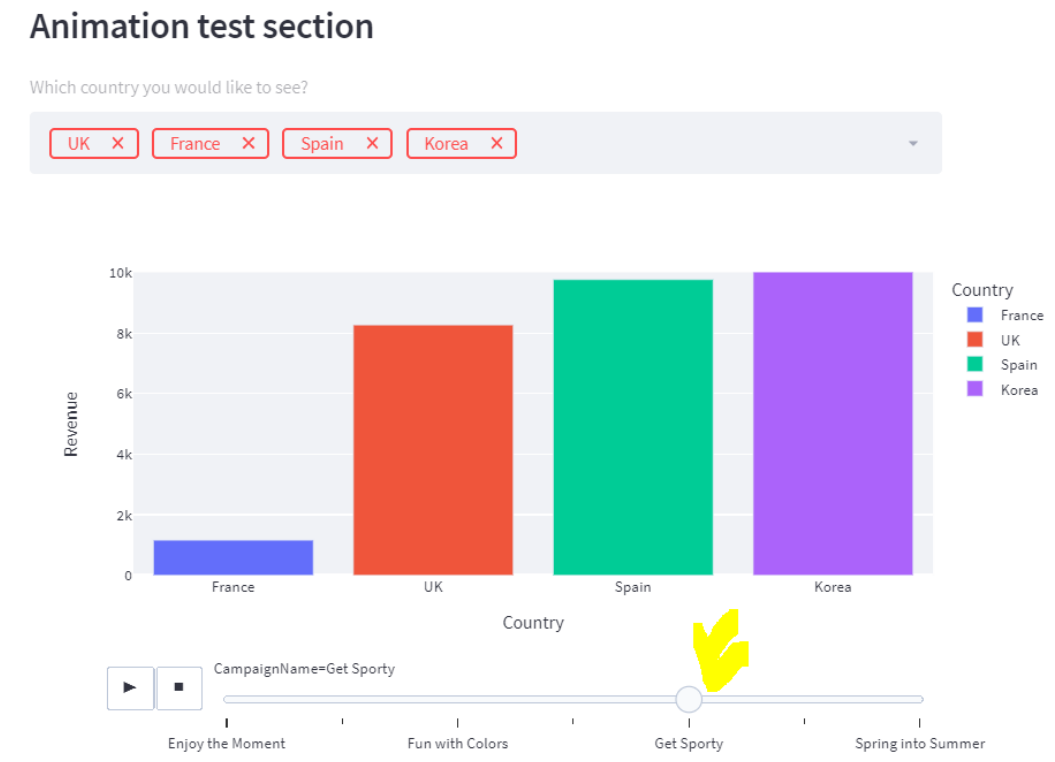
We also have added one more scatter plot for the revenue. On the left side of the page we have the filters and we can select the range for the revenue and display the results in a scatterplot.



An extra thing we tried to experiment for this project is the animation. As you can see below the user is able to select the country and we have made it to group by campaign Name. Based on the country and the campaign name the user will see how the revenue changes.

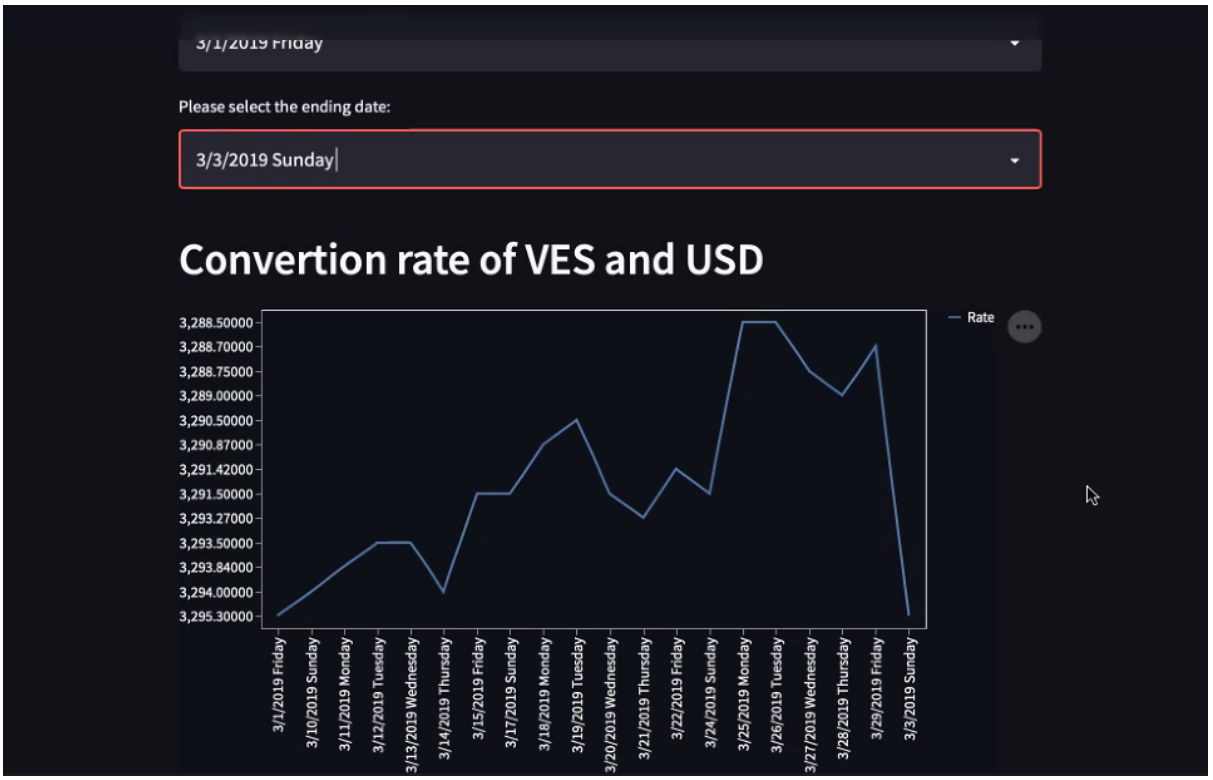


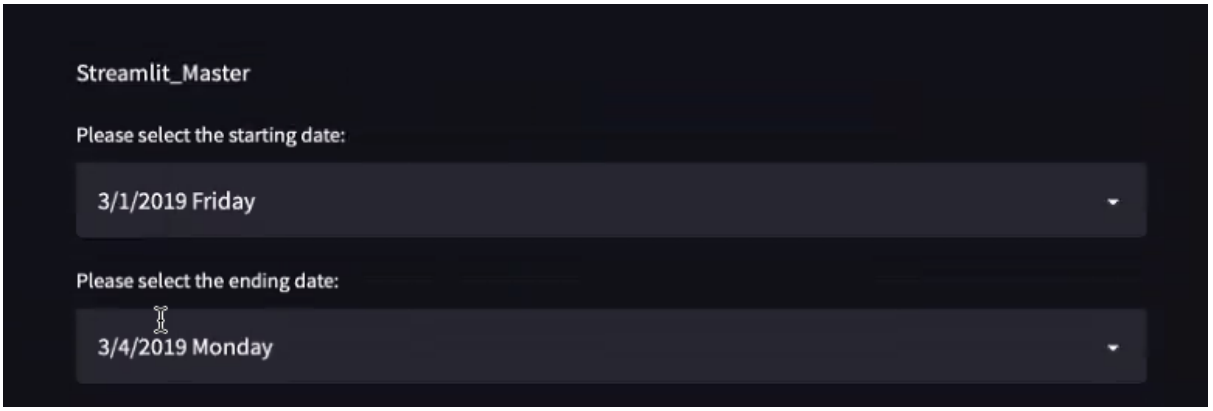
If the use slides through the other campaign this is how the graph will change:

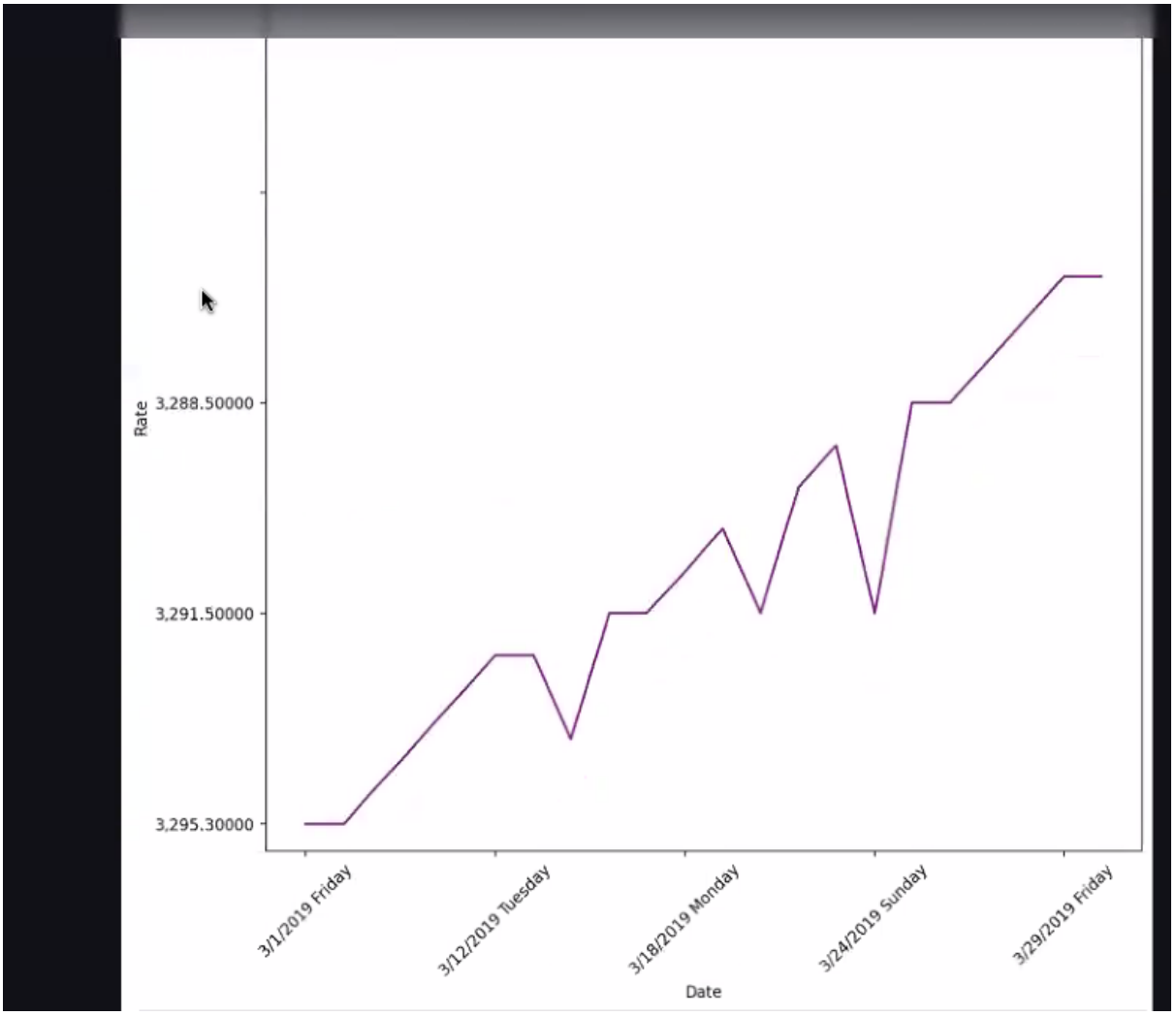


Extra bonus tasks(requirements taken in the last day of the project) :

Task 1
Load the file ves-usd.csv from the ML-datasets repository: https://github.com/matzim95/ML-datasets and plot a linear graph to show the dependence between bolivar and US dollar.





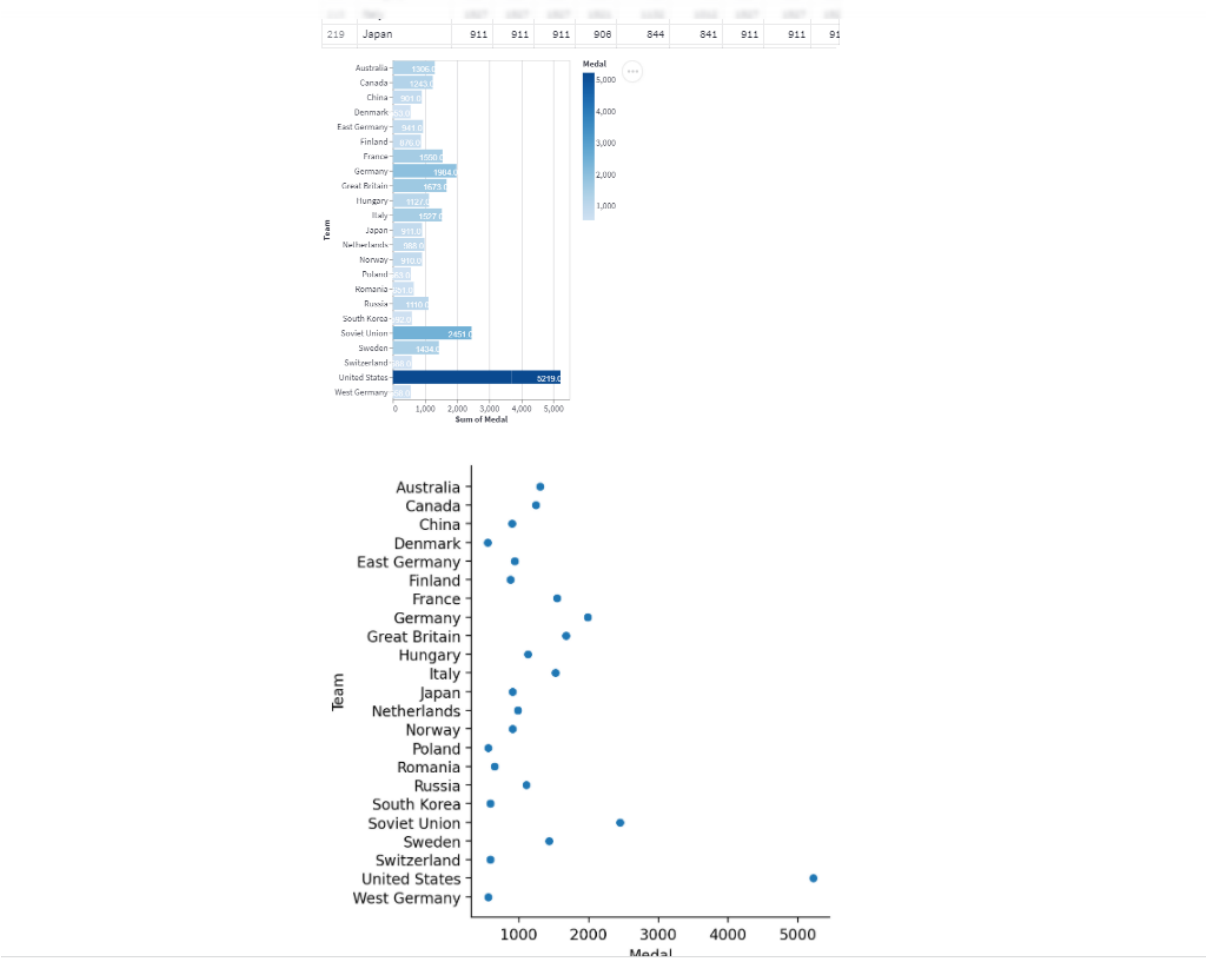


Task 3

Analyze the data from Olympic Games (look into the file olympic.csv https://github.com/matzim95/ML-datasets) and make a bar chart showing the number of medals the country has won. Focus only on these countries that got more than 500 medals in total. Use the colors to indicate whether the medal was gold, silver or bronze.

Task 4

Using the set from the previous task, create a scatter plot showing the relationship between the height, the weight, and the number of medals that were won by players at the Olympics. Focus only on these players that got more than 5 medals in total.



Task 5:
Load a co2 table from the repository and display the scatter plot showing average change of CO2 emission over time using seaborn. Think about odd values that could be mistakes and how we can filter them out. Ignore Trend and Interpolated columns.

Task 6

Make a boxplot, a violinplot and swarmplot plots for Age, Height and Weight of players from one kind of sports from the Olympians dataset using the seaborn library.

The Olympic Dataset

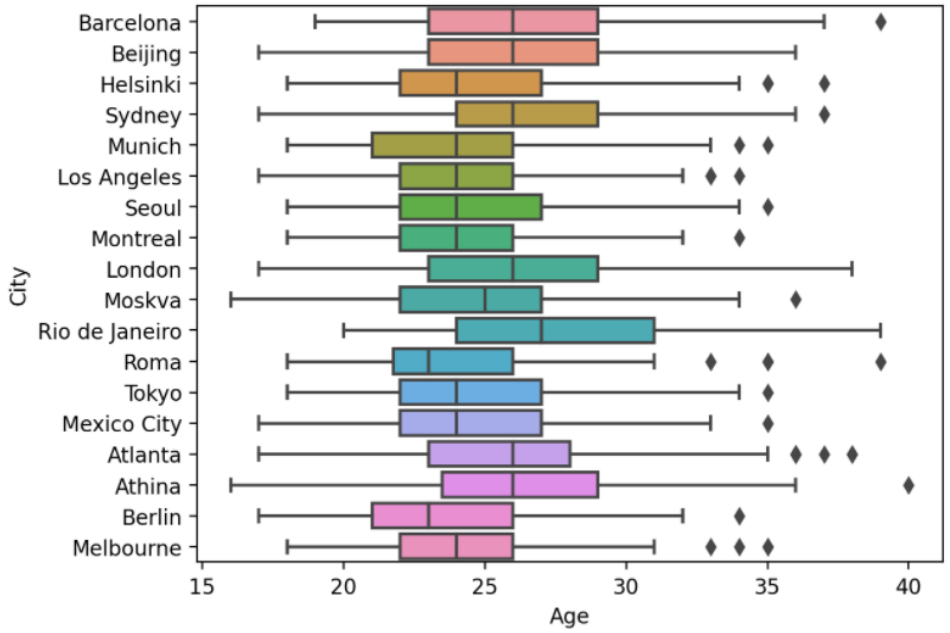
	ID	Name	Sex	Age	Height	Weight	Team
0	1	A Dijiang	M	24.0000	180.0000	80.0000	China
1	2	A Lamusi	M	23.0000	170.0000	60.0000	China
2	3	Gunnar Nielsen Aaby	M	24.0000	<NA>	<NA>	Denma
3	4	Edgar Lindenau Aabye	M	34.0000	<NA>	<NA>	Denma
4	5	Christine Jacoba Aaftink	F	21.0000	185.0000	82.0000	Netherl
5	5	Christine Jacoba Aaftink	F	21.0000	185.0000	82.0000	Netherl
6	5	Christine Jacoba Aaftink	F	25.0000	185.0000	82.0000	Netherl
7	5	Christine Jacoba Aaftink	F	25.0000	185.0000	82.0000	Netherl
8	5	Christine Jacoba Aaftink	F	27.0000	185.0000	82.0000	Netherl
9	5	Christine Jacoba Aaftink	F	27.0000	185.0000	82.0000	Nether

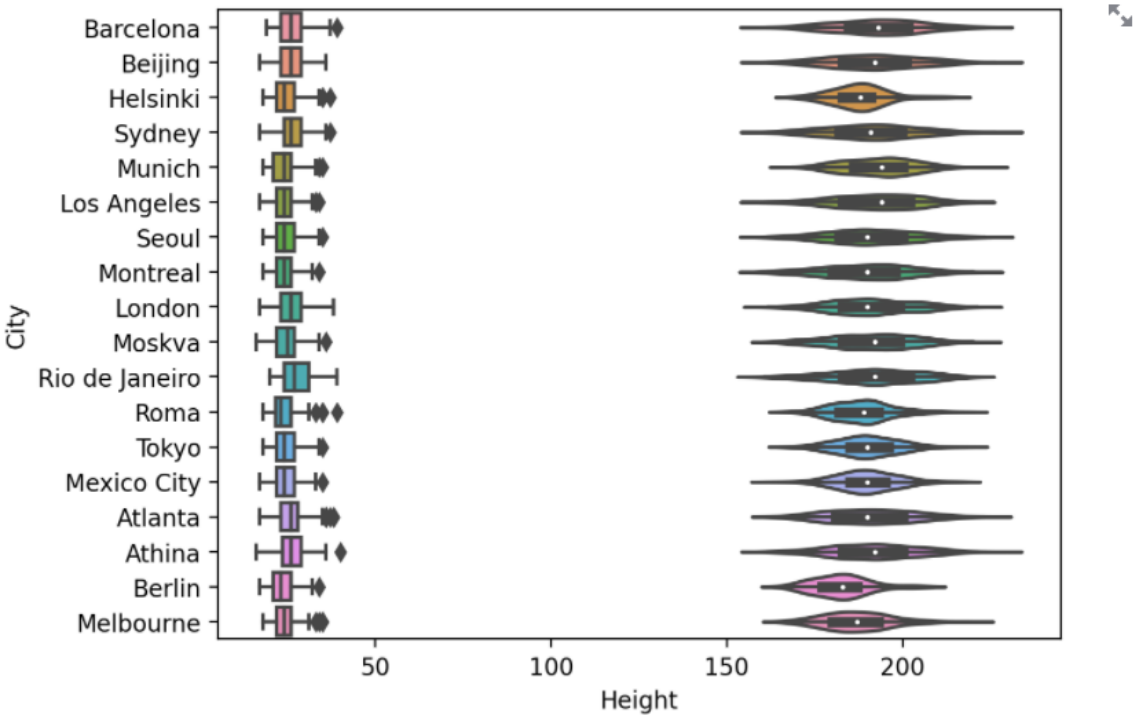
Boxplot for a sport based on Age, Height, Weight

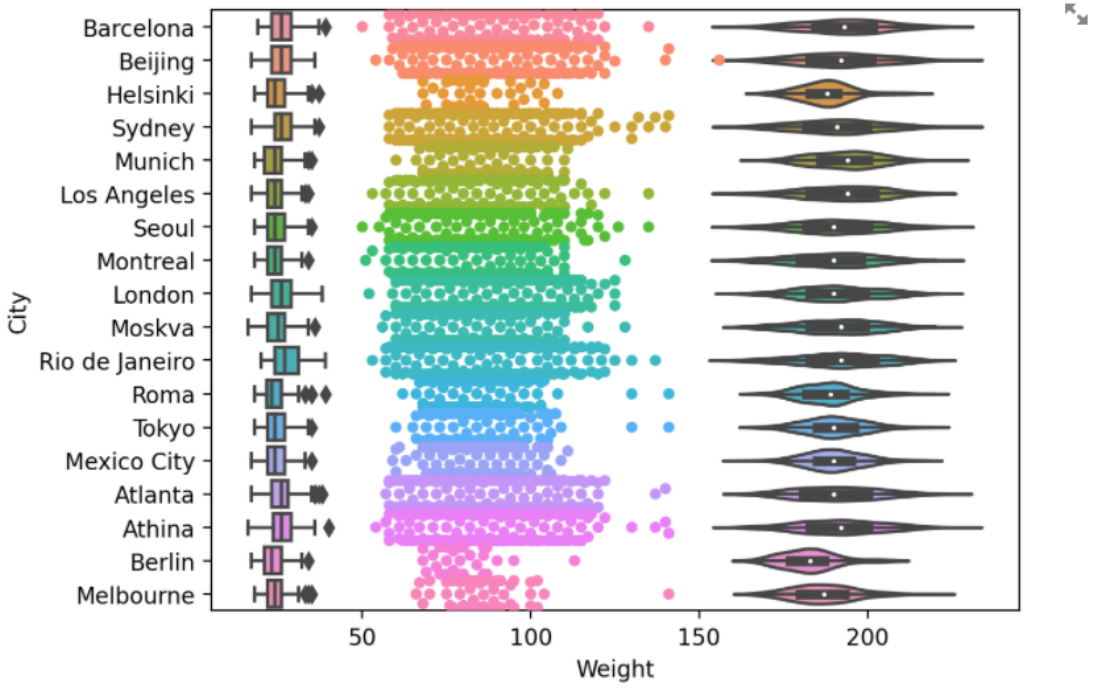
Please select the sport:

Basketball

		Year	Season	City	Sport	Event	Medal
0	r	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	<NA>
167	r	2008	Summer	Beijing	Basketball	Basketball Women's Bask…	<NA>
250	r	1952	Summer	Helsinki	Basketball	Basketball Men's Basketball	<NA>
264	r	2000	Summer	Sydney	Basketball	Basketball Men's Basketball	<NA>
346	r	1972	Summer	Munich	Basketball	Basketball Men's Basketball	<NA>
359	r	1984	Summer	Los Angeles	Basketball	Basketball Men's Basketball	<NA>
360	r	1988	Summer	Seoul	Basketball	Basketball Men's Basketball	<NA>
363	r	1972	Summer	Munich	Basketball	Basketball Men's Basketball	<NA>
364	r	1976	Summer	Montreal	Basketball	Basketball Men's Basketball	<NA>
490	r	1988	Summer	Seoul	Basketball	Basketball Men's Basketball	<NA>


Barcelona
Beijing
Helsinki
Sydney
Munich
Los Angeles
Seoul
Montreal
London
Moskva
Rio de Janeiro
Roma
Tokyo
Mexico City
Atlanta
Athina
Berlin
Melbourne
City
15
20
25
30
35
40
Age


Barcelona
Beijing
Helsinki
Sydney
Munich
Los Angeles
Seoul
Montreal
London
Moskva
Rio de Janeiro
Roma
Tokyo
Mexico City
Atlanta
Athina
Berlin
Melbourne
City
50
100
150
200
Height


Barcelona
Beijing
Helsinki
Sydney
Munich
Los Angeles
Seoul
Montreal
London
Moskva
Rio de Janeiro
Roma
Tokyo
Mexico City
Atlanta
Athina
Berlin
Melbourne
City
50
100
150
200
Weight