😊 Mapping Data Flow – Fact Table

Mapping Data Flow in Azure Data Factory is a visually based data transformation tool that allows you to design and execute data transformation processes within Azure Data Factory. It provides a graphical interface for building data transformation logic without writing code, making it easier for users to manipulate and transform data at scale.

Key features of Mapping Data Flow include:

1. **Visual Interface:** It offers a drag-and-drop interface where users can visually design data transformation processes by connecting various data sources, transformations, and sinks.
2. **Built-in Transformations:** Mapping Data Flow provides a wide range of built-in transformations such as filtering, sorting, aggregating, joining, and pivoting, among others, allowing users to perform complex data transformations without coding.
3. **Data Preview and Debugging:** Users can preview the data at each step of the transformation process and debug their data flows within the visual interface, making it easier to identify and fix issues.
4. **Integration with Azure Services:** Mapping Data Flow seamlessly integrates with other Azure services such as Azure Data Lake Storage, Azure Synapse Analytics, and Azure SQL Database, allowing users to easily ingest and transform data from various sources.
5. **Auto-scaling:** It automatically scales resources based on the data volume and complexity of transformations, ensuring optimal performance and cost efficiency.
6. **Monitoring and Management:** Azure Data Factory provides monitoring and management capabilities for Mapping Data Flows, allowing users to monitor the execution of data flows, track performance metrics, and manage dependencies between different data flows.

😄 Use cases of Mapping data flow:

Mapping Data Flow in Azure Data Factory can be used in various scenarios across industries for data transformation and processing. Here are some common use cases:

1. **ETL (Extract, Transform, Load) Pipelines:** Mapping Data Flow can be used to build ETL pipelines for ingesting data from various sources, performing transformations such as cleansing, deduplication, and enrichment, and loading the transformed data into a target data store or data warehouse.
2. **Data Warehousing:** Organizations can use Mapping Data Flow to transform and load data into data warehouses such as Azure Synapse Analytics (formerly Azure SQL Data Warehouse). This includes tasks like aggregating, joining, and filtering data before loading it into the warehouse for analytics and reporting.
3. **Data Integration:** Mapping Data Flow can be used to integrate data from disparate sources, such as databases, files, and streaming data sources, into a unified format. This allows organizations to create a single source of truth for their data and enable better decision-making.

4. **Data Migration:** During data migration projects, Mapping Data Flow can be used to transform data from legacy systems or on-premises databases to cloud-based data stores or modern data platforms. It facilitates the mapping of data structures and transformations required for the migration process.
5. **Real-time Data Processing:** For real-time analytics and processing, Mapping Data Flow can be integrated with streaming data sources such as Azure Event Hubs or Apache Kafka. It enables organizations to perform real-time transformations on streaming data streams before storing or analyzing them.
6. **Data Quality Management:** Mapping Data Flow can be used to implement data quality checks and transformations to ensure that data is accurate, consistent, and compliant with business rules and regulations. This includes tasks such as data validation, standardization, and error handling.
7. **Machine Learning Data Preparation:** In machine learning projects, Mapping Data Flow can be used for data preparation tasks such as feature engineering, data normalization, and feature selection. It helps data scientists to prepare clean and structured datasets for training machine learning models.
8. **Customer 360 View:** Organizations can use Mapping Data Flow to create a unified view of customer data by integrating and transforming data from multiple sources such as CRM systems, transactional databases, and social media platforms. This enables organizations to gain insights into customer behavior and preferences.

In this guide, we are leveraging Mapping Data Flow within Azure Data Factory to orchestrate a data transformation pipeline. Our objective is to seamlessly integrate data from disparate sources, perform necessary transformations like joining and filtering, and then load the refined data into a target destination, such as a data warehouse or database. By utilizing Mapping Data Flow's visual interface and built-in transformations, we simplify the complex process of data transformation without the need for extensive coding. The end goal is to establish a streamlined data processing workflow that enhances data quality, enables more efficient data analysis, and facilitates informed decision-making within organizations. Through this approach, we aim to maximize the value of data assets by ensuring they are accurate, consistent, and actionable for business insights.

😄 To begin with the Lab:

The below script is a SQL query that retrieves data from two tables: [SalesLT].[SalesOrderDetail] (aliased as dt) and [SalesLT].[SalesOrderHeader] (aliased as hd). It performs a left join between these tables based on the SalesOrderID column, ensuring that all records from the left table (SalesOrderDetail) are included in the result set, regardless of whether there is a matching record in the right table (SalesOrderHeader).

Here's a breakdown of the query:

- SELECT dt.[ProductID], dt.[SalesOrderID], dt.[OrderQty], dt.[UnitPrice], hd.[OrderDate], hd.[CustomerID], hd.[TaxAmt]: This part of the query selects specific columns from both tables. Columns selected include ProductID, SalesOrderID,

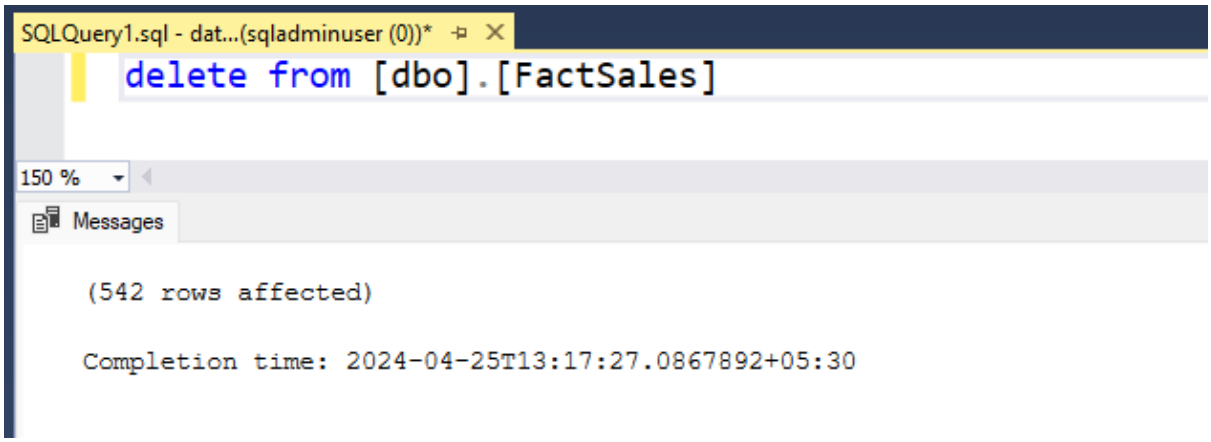
OrderQty, and UnitPrice from the SalesOrderDetail table (dt alias), and OrderDate, CustomerID, and TaxAmt from the SalesOrderHeader table (hd alias).

- FROM [SalesLT].[SalesOrderDetail] dt: This part specifies the source table SalesOrderDetail and aliases it as dt.
- LEFT JOIN [SalesLT].[SalesOrderHeader] hd ON dt.[SalesOrderID] = hd.[SalesOrderID]: This part performs a left join between the SalesOrderDetail table (dt) and the SalesOrderHeader table (hd) based on the SalesOrderID column. It ensures that all records from the SalesOrderDetail table are included in the result set, even if there is no matching SalesOrderID in the SalesOrderHeader table.
- Overall, this query retrieves data related to sales orders, including product details (ProductID, OrderQty, UnitPrice) from the SalesOrderDetail table and order information (OrderDate, CustomerID, TaxAmt) from the SalesOrderHeader table. It's useful for analyzing sales data and customer purchase behavior within the specified database schema.

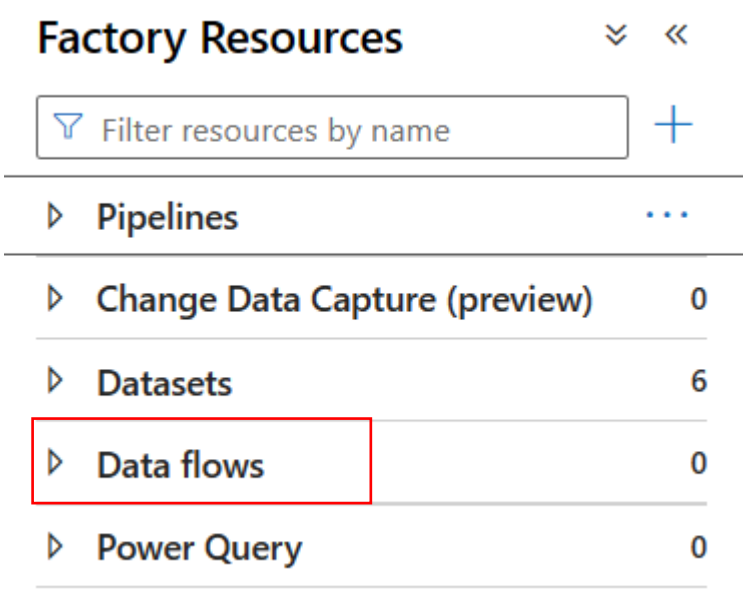
```
SELECT
dt.[ProductID],dt.[SalesOrderID],dt.[OrderQty],dt.[UnitPrice],hd.[OrderDate],hd.[CustomerID],hd.[TaxAmt]

 FROM [SalesLT].[SalesOrderDetail] dt

 LEFT JOIN [SalesLT].[SalesOrderHeader] hd

 ON dt.[SalesOrderID]=hd.[SalesOrderID]
```

---

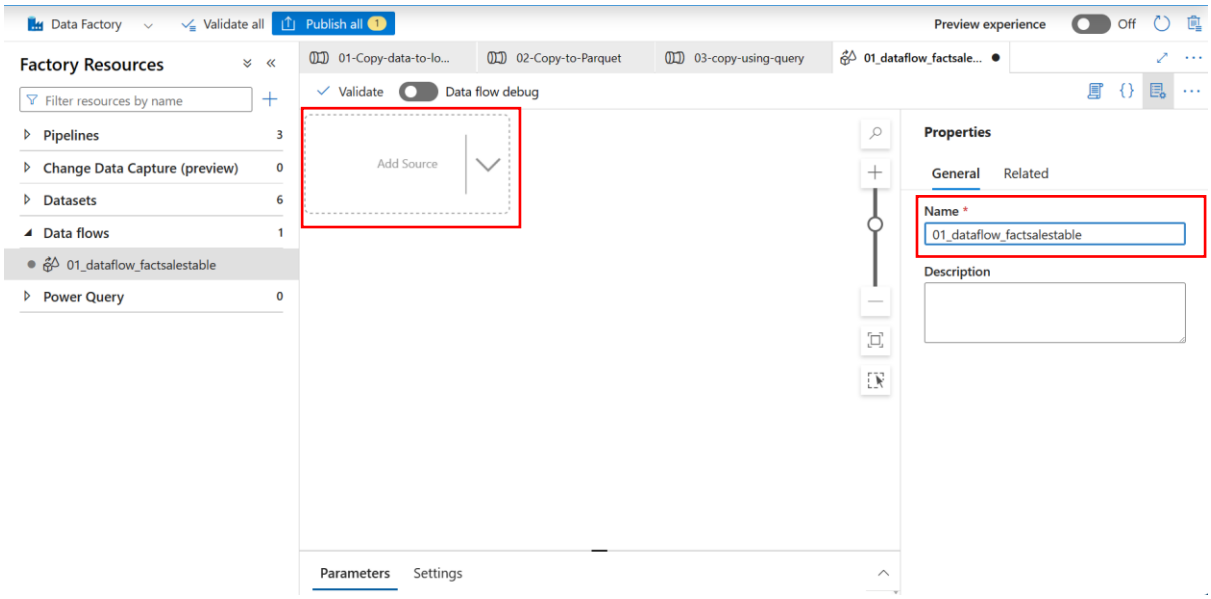
1. In the lab we will make use of mapping data flow and through this, we will create tables.
2. Now go to your SSMS and delete the data from your fact sales table.



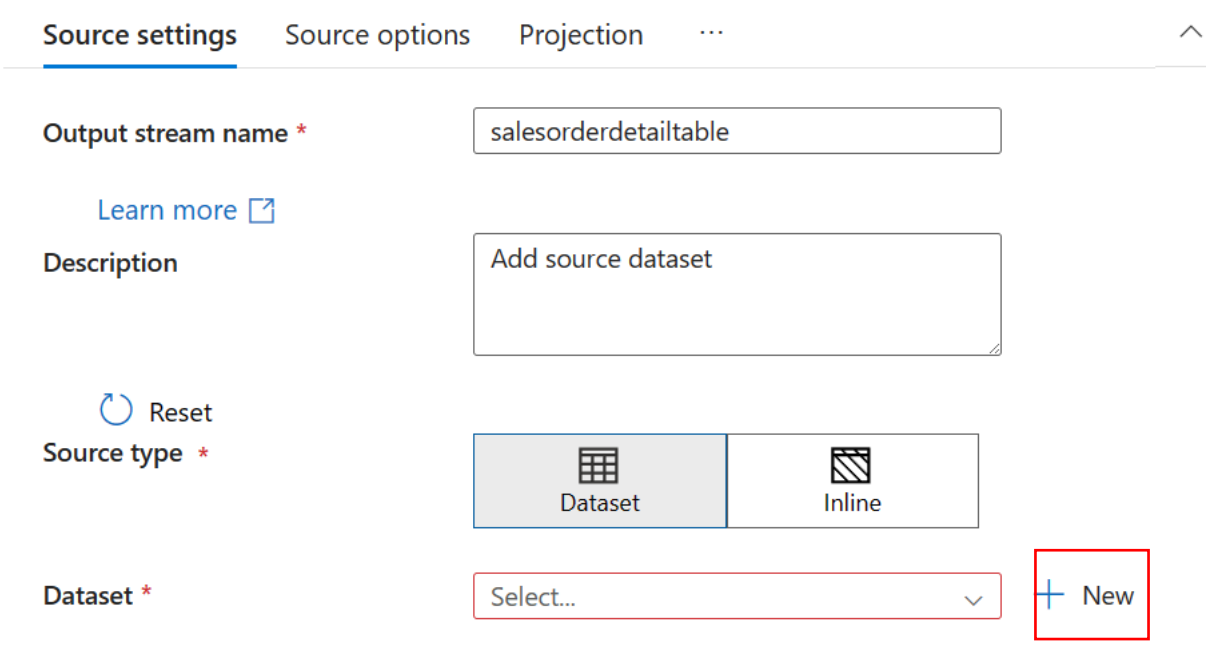
3. To build a Mapping data flow, you need to navigate to the Data Factory wizard.
4. From the factory resources you need to click on data flows and select to build a new data flow.



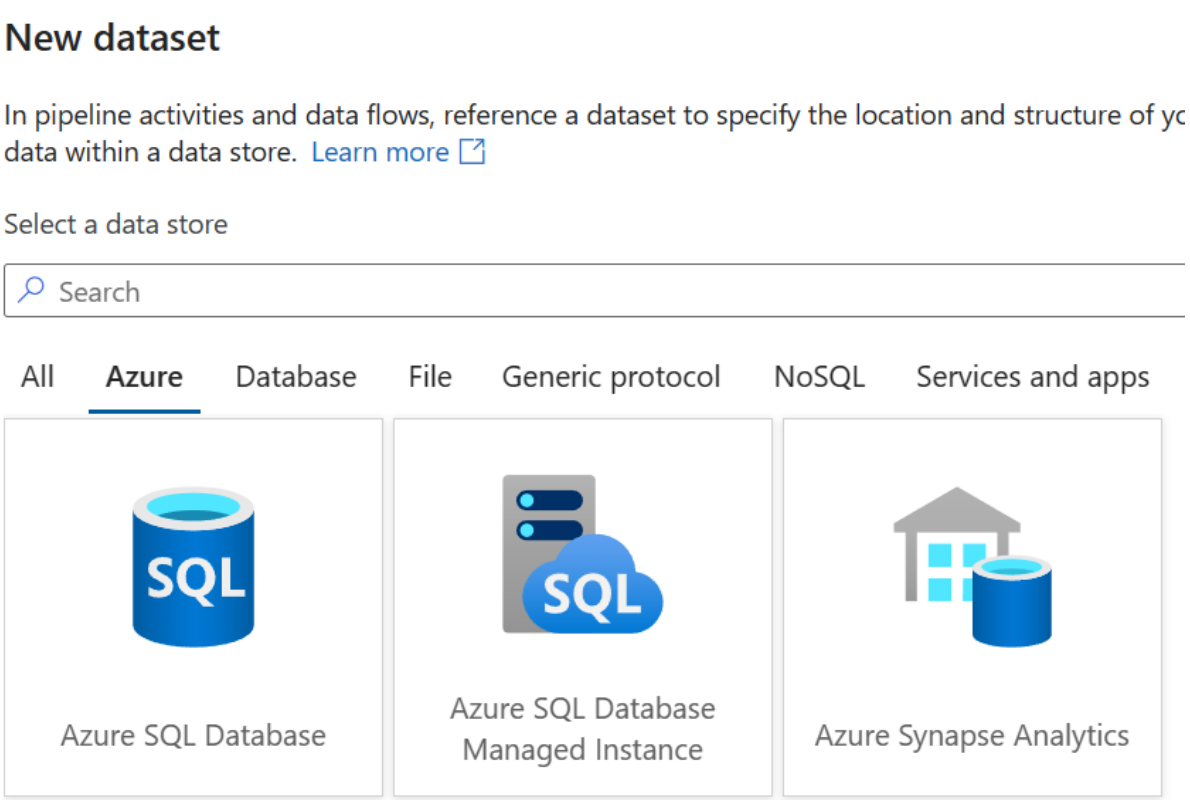
5. Then the wizard will give you this type of interface. Here you need to give your data flow a name.
6. After that you need to click on add source and add your first source.



7. Now you have to select your first source which will sales order detail table.
8. First you are going to give a name to your output stream.
9. Then you are going to add a data set to it which will be your SQL Database because we are taking the data from it.
10. Now click on new to add dataset.



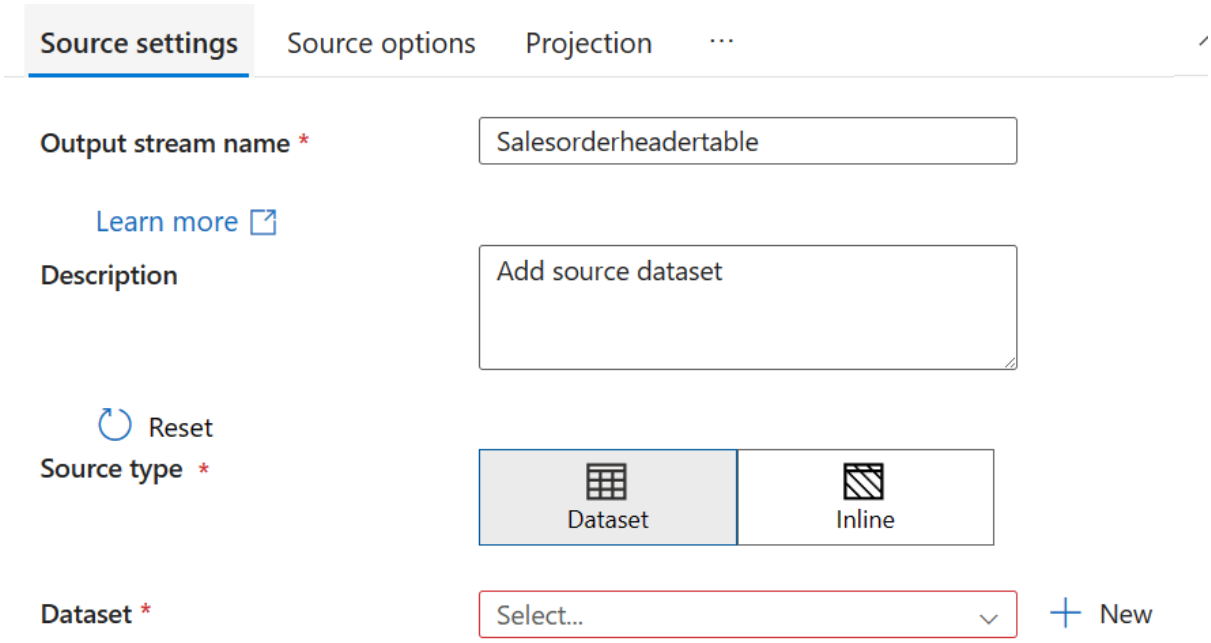
11. Then you need to add your Azure SQL Database.



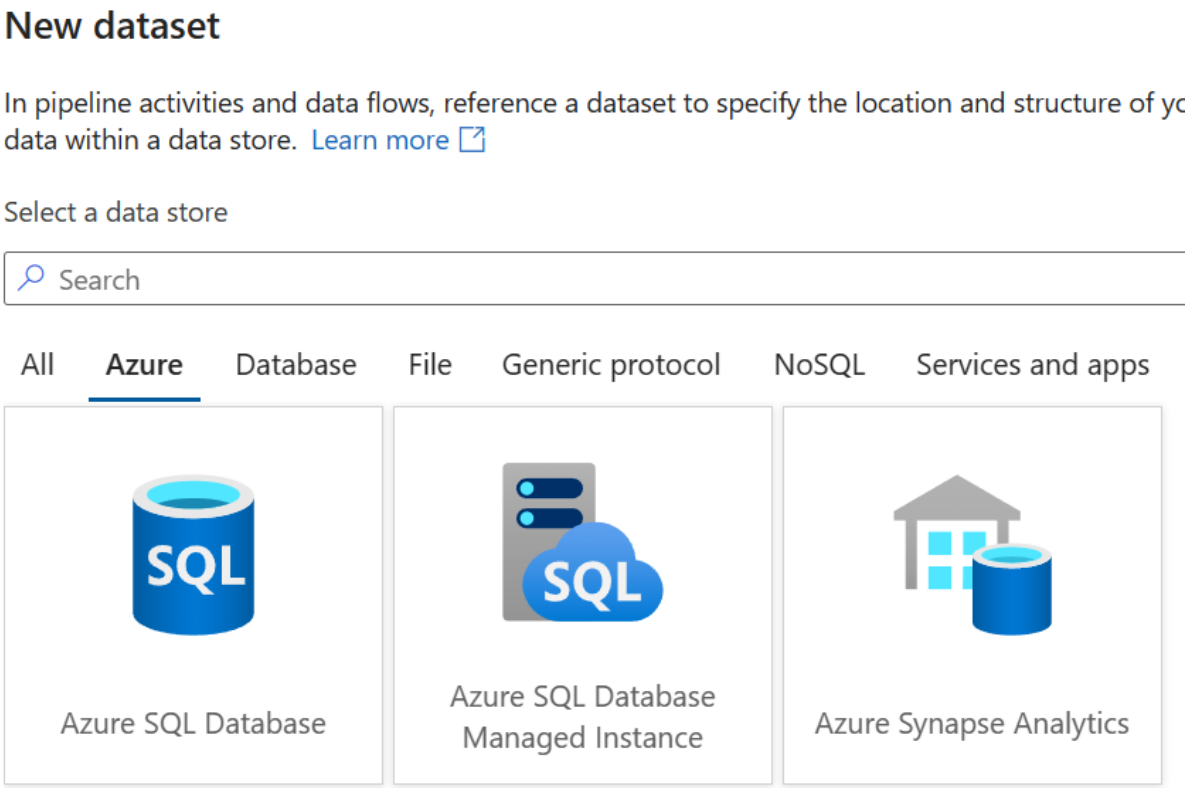
12. In the set properties, first you need to give it a name after that choose your linked service. Then choose your table name as shown below. Then just click on OK.
13. So please remember here, we are in something known as a mapping data flow in which we are building a visual representation of the data flow itself.

14. After adding your first source then we are going to add another source from the canvas like we did for the first time. Again click on add source and add your second source.

15. First you need to give your output stream a name, then add your dataset for the sales order header table. For that again click on new.



16. Then again you have to choose Azure SQL Database.



17. Then in the properties, first give it a name then choose your linked service. After that you have choose the table name as shown below then just click on OK.

Set properties

Name

Salesorderheader_table

Linked service *

demodb_service

Table name

SalesLT.SalesOrderHeader

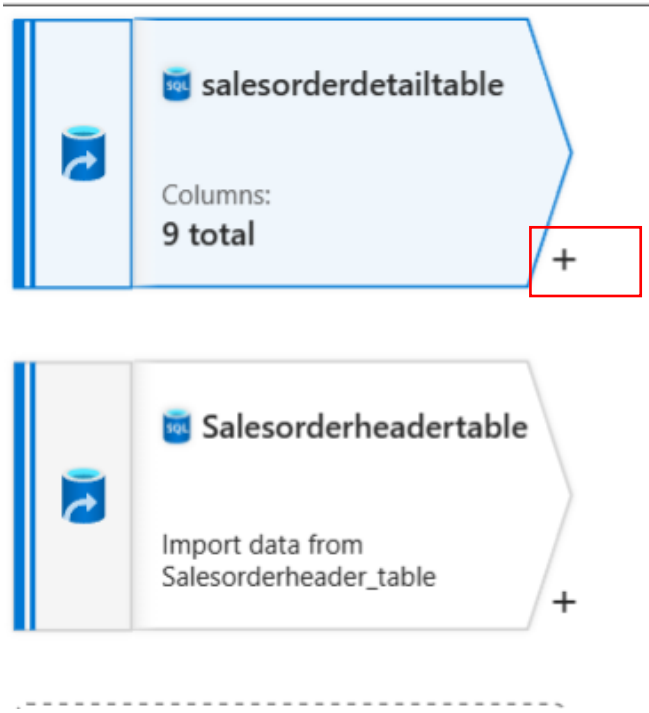
Enter manually

Import schema

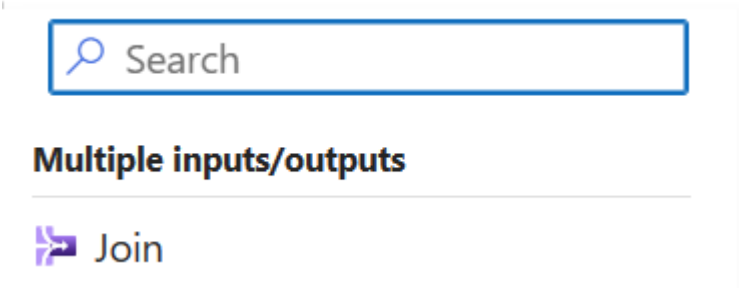
From connection/store None

Advanced

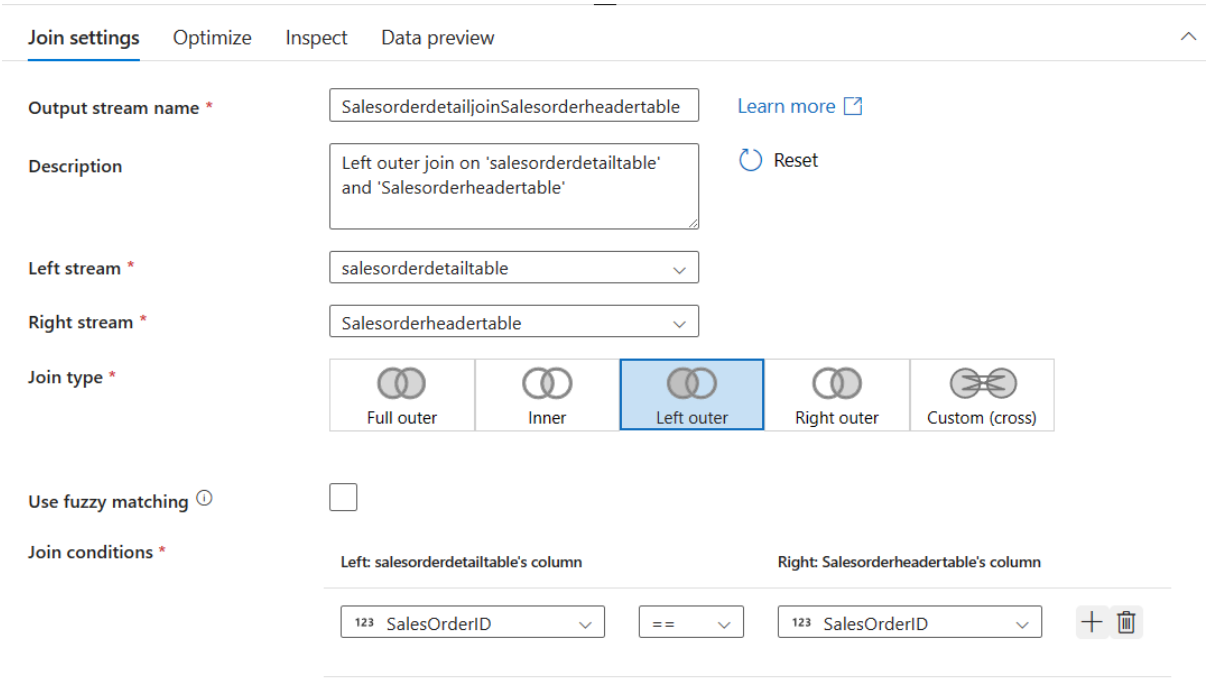
18. Until now we have created two streams, and now we are going to join them. For that, you need to click on this highlighted plus sign from the first source as shown below.



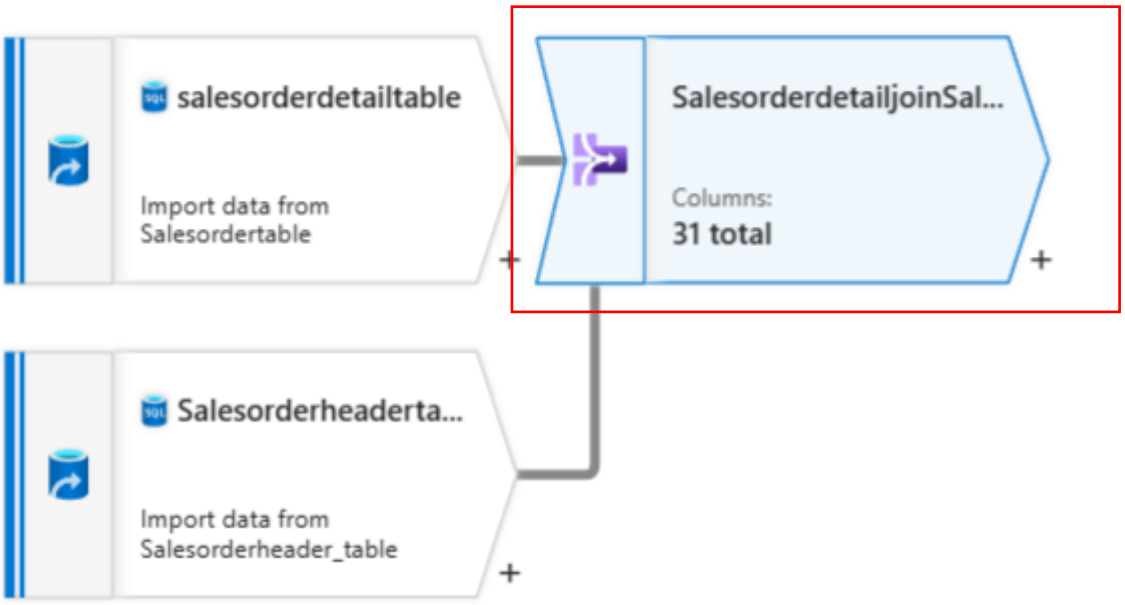
19. Then you have to choose the join option from the drop-down menu. Now with this option, we can join multiple streams of data.



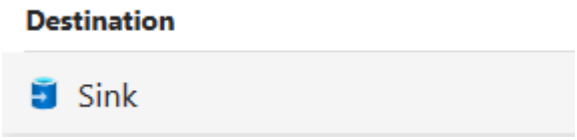
20. Then for the join settings, first you need to give it a name then in the left stream choose your sales order detail table and in the right stream choose sales order header table.
21. After that the join type should be Left outer and in the Join condition choose Sales Order ID for both sides.



22. So, this highlighted part is your join for both of the tables. Now you have to click on the plus sign from join and choose sink option from the drop-down menu.

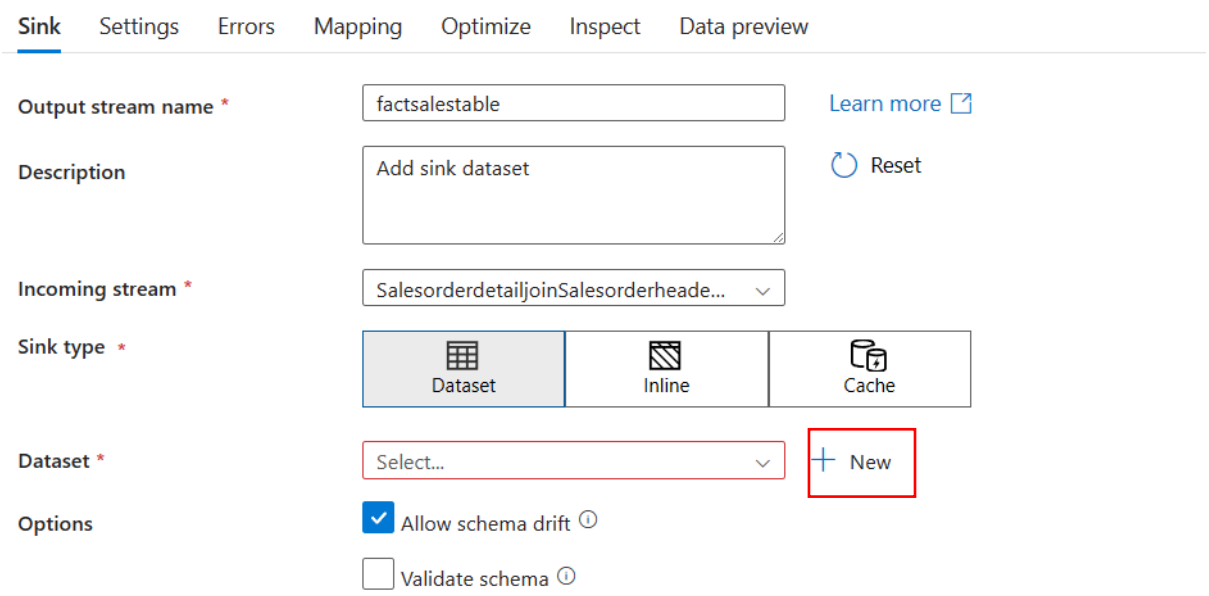


23. This sink option is for the destination and if you will scroll down to the bottom you will find it.

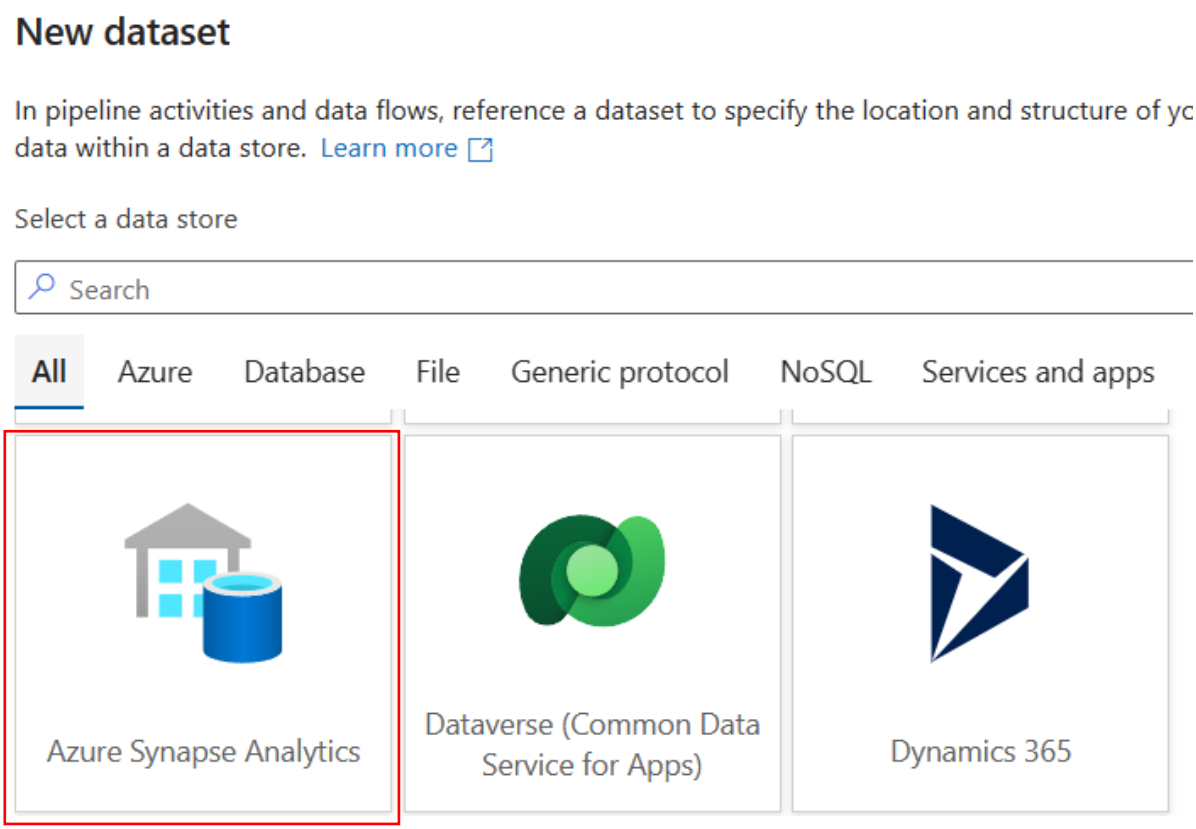


24. Below you can that the sink option or rather say the destination option is connected.

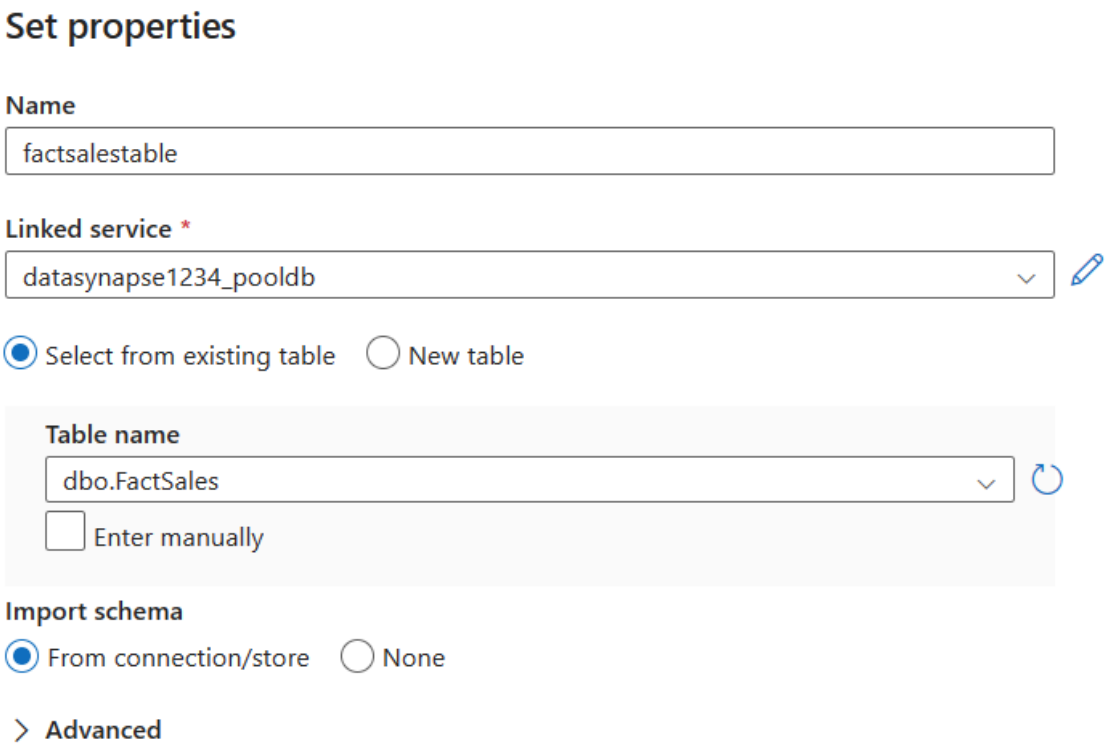
25. Now give your sink a name then you have to choose Azure Synapse as your Dataset. For that click on New.



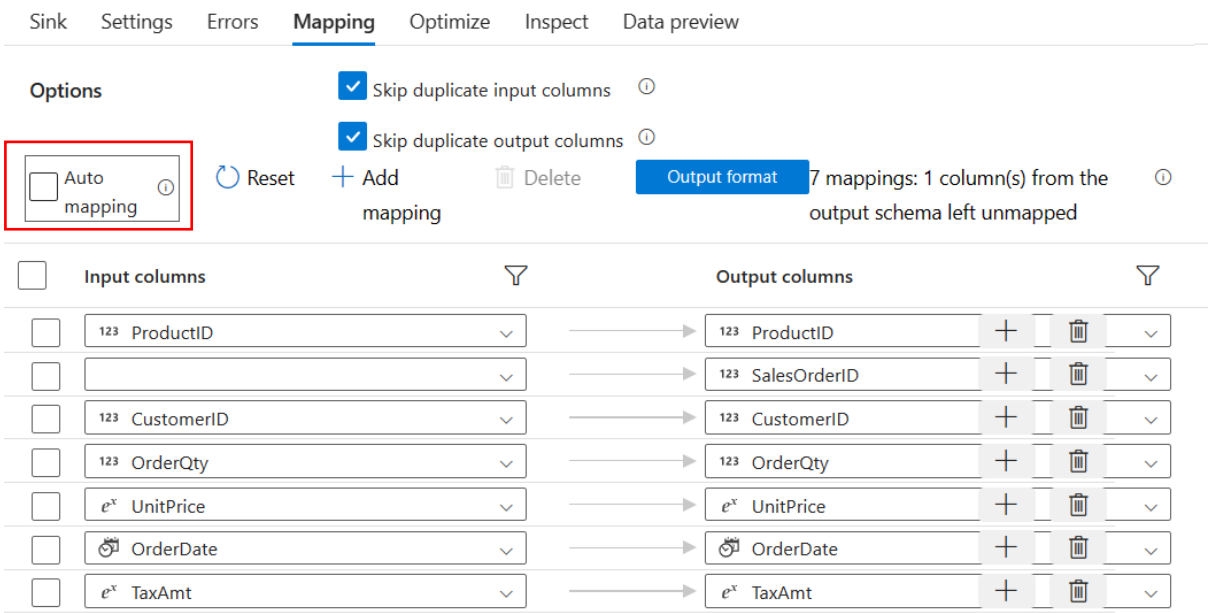
26. Then choose Azure Synapse Analytics and click on continue.



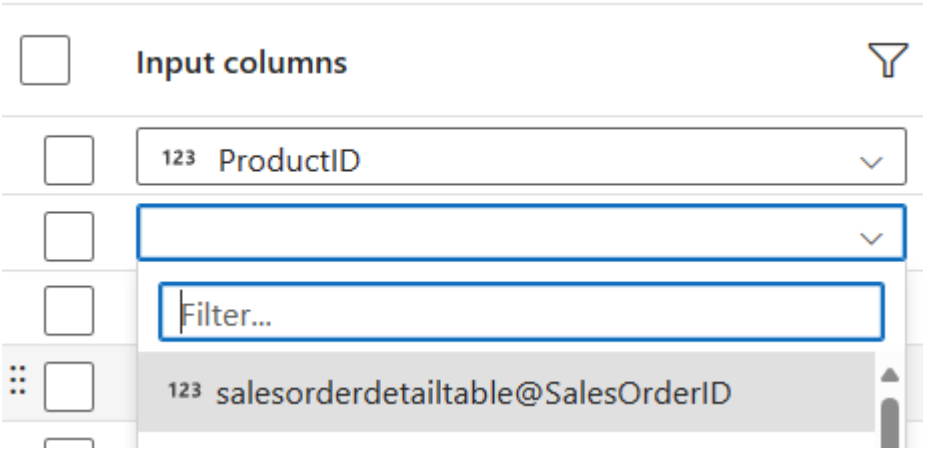
27. Now to set its properties, first give it a name then choose your linked service.
28. After that you have to choose the table as shown below then click on OK.



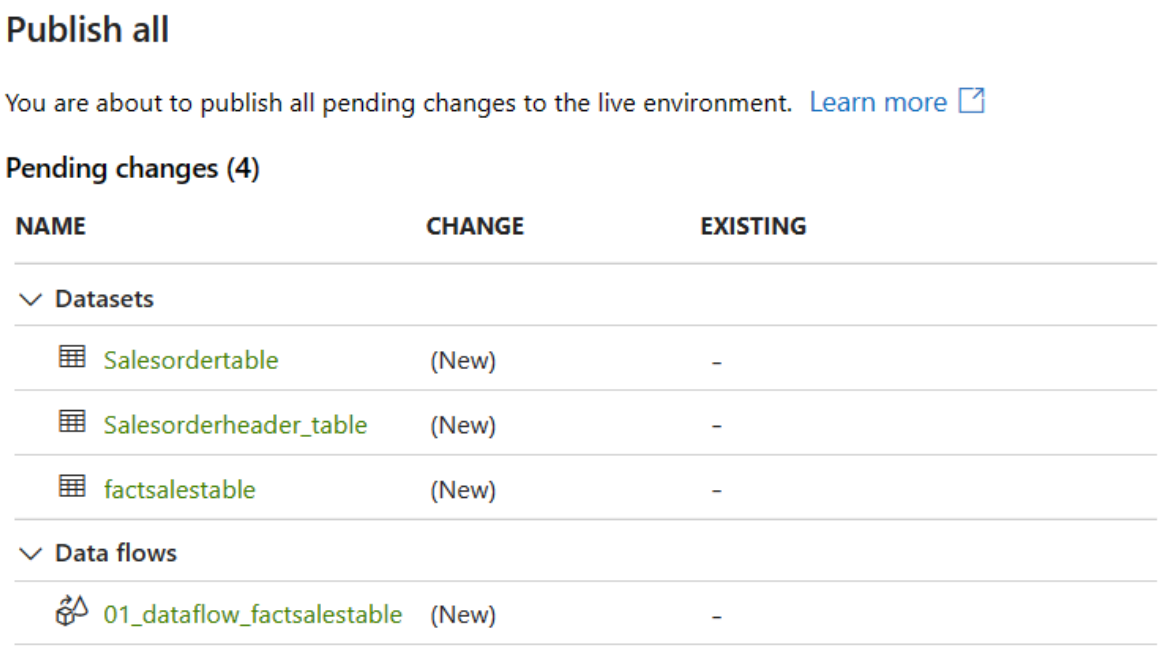
29. After that go to mappings and uncheck the option for auto mapping. Here you will see that the Sales Order ID has not been matched yet because we have Sales Order ID on both the tables.
30. We need to choose this ourselves.



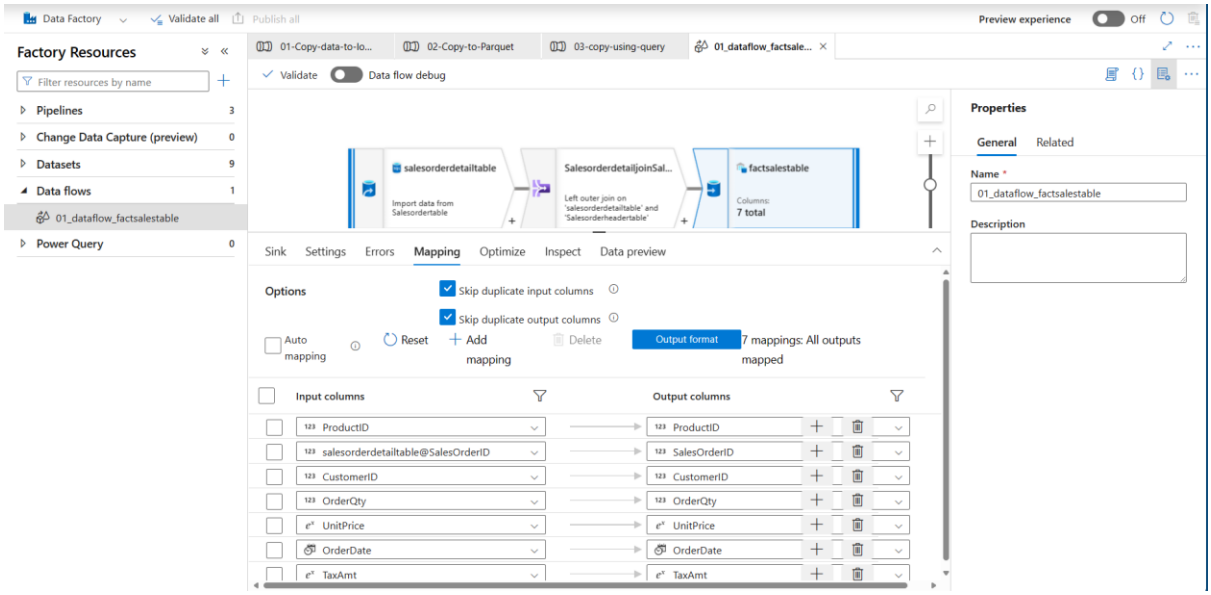
31. While mapping, choose this one shown below as it is from your sales order detail table.



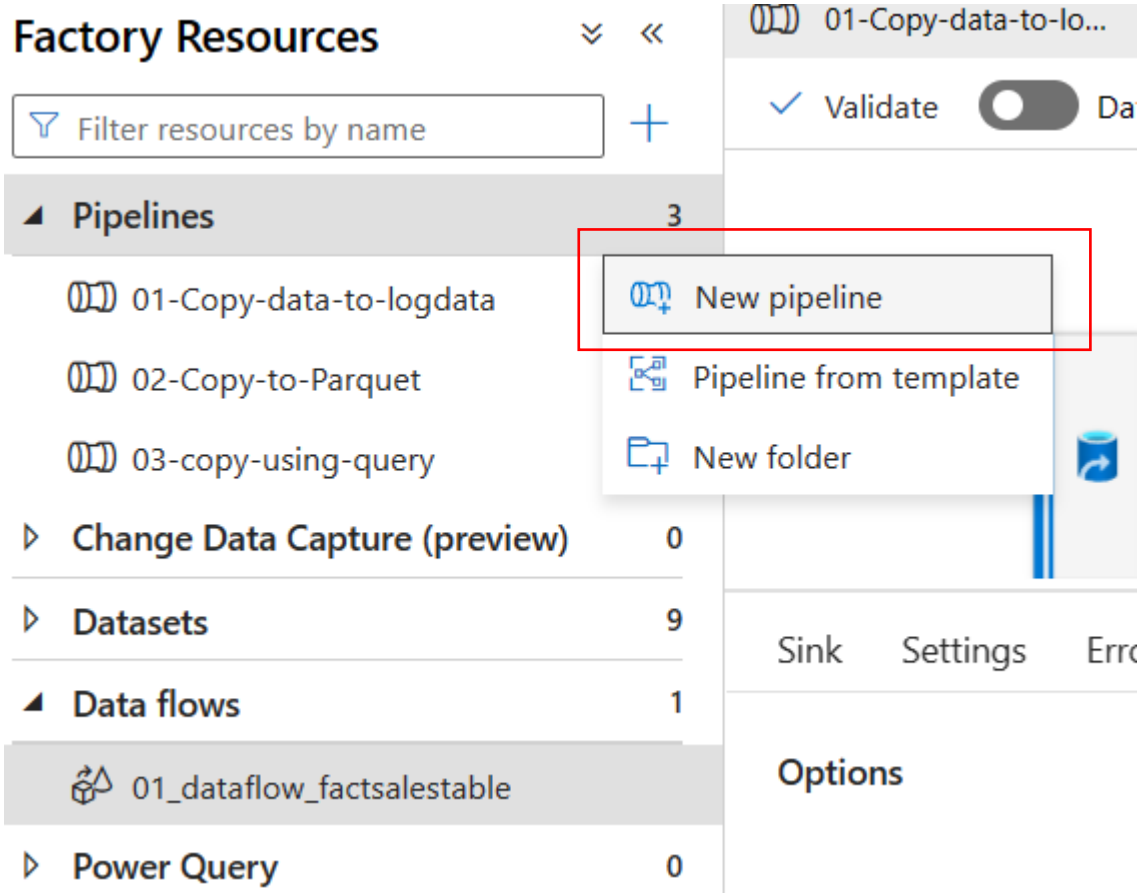
32. After that we will click on validate all and then click on publish all. This will publish all your resources.



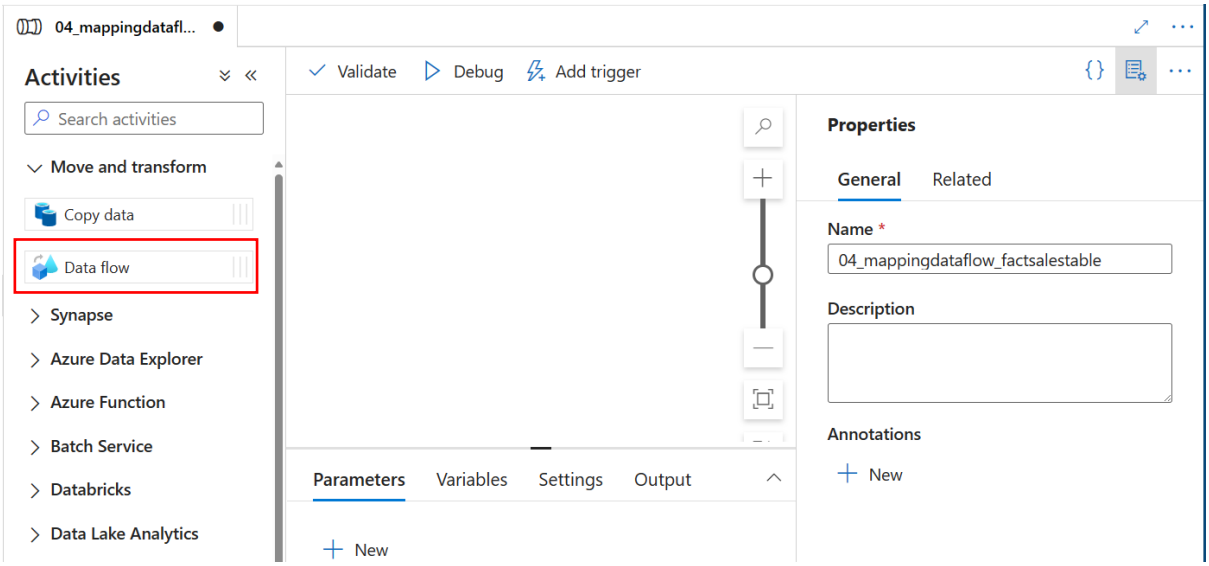
33. After publishing all your data you will see that we do not have the option to trigger this or in simple words, we cannot run it.
34. So, here if you will see, we just created a mapping data flow wherein we just mapped our data nothing more.



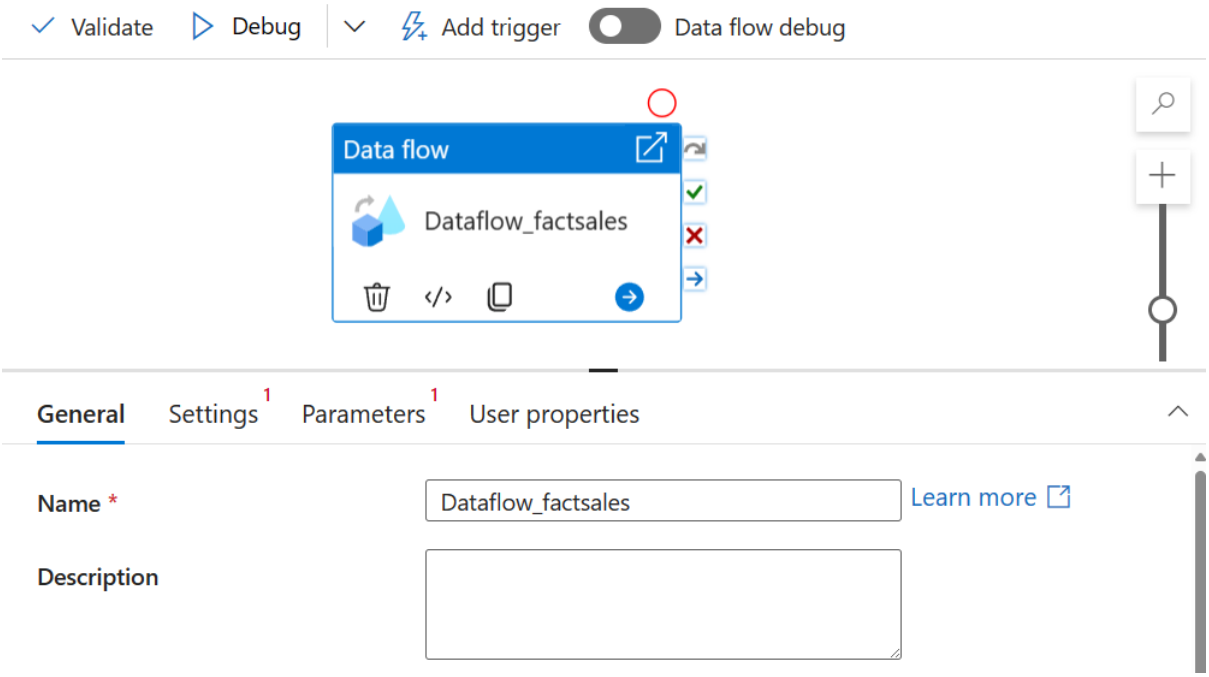
35. So, for that we are going to create a pipeline. Now from factory resources choose a pipeline and create a new pipeline.



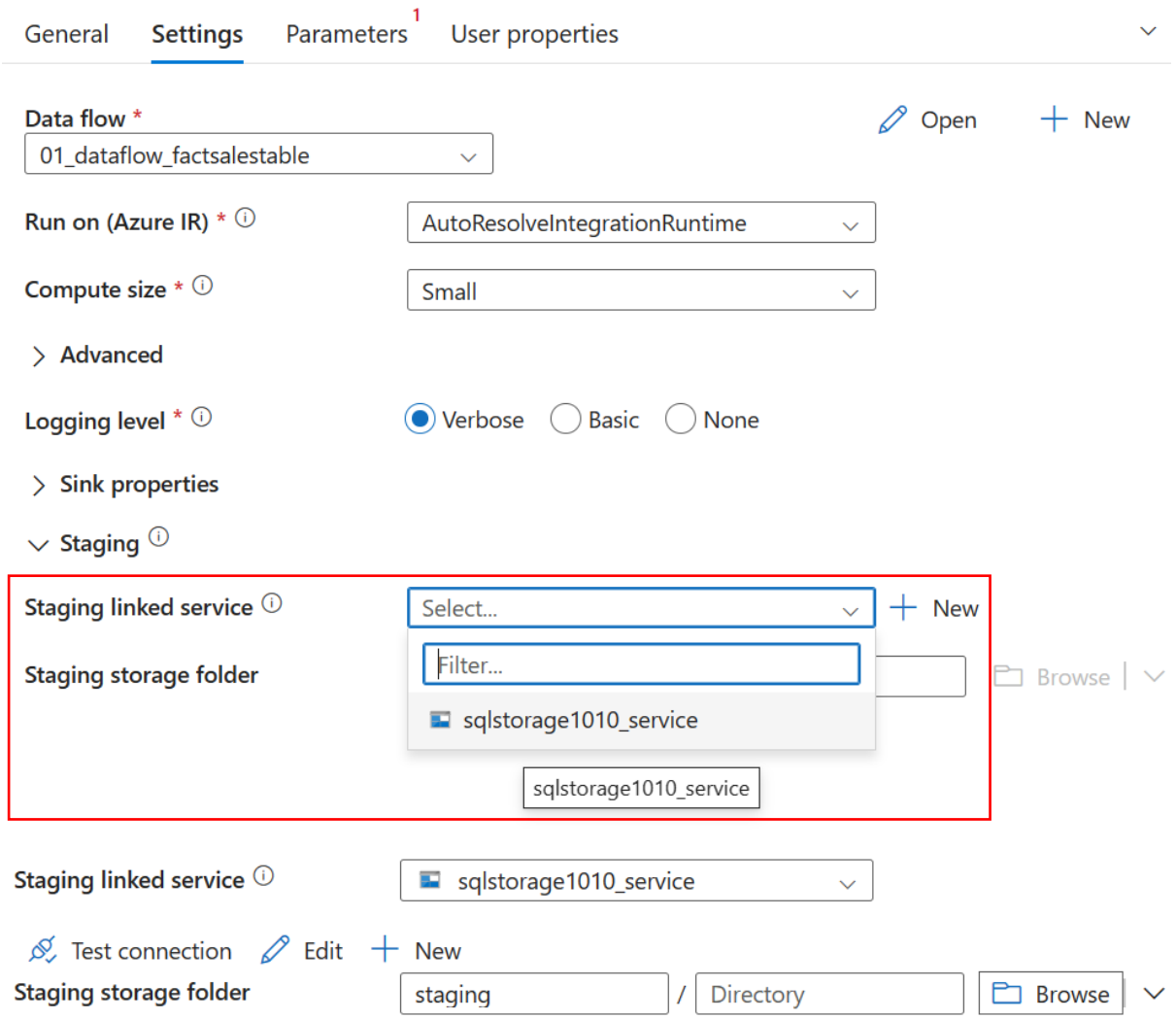
36. Here first we are going to give our pipeline a name then drag the data flow to the canvas.



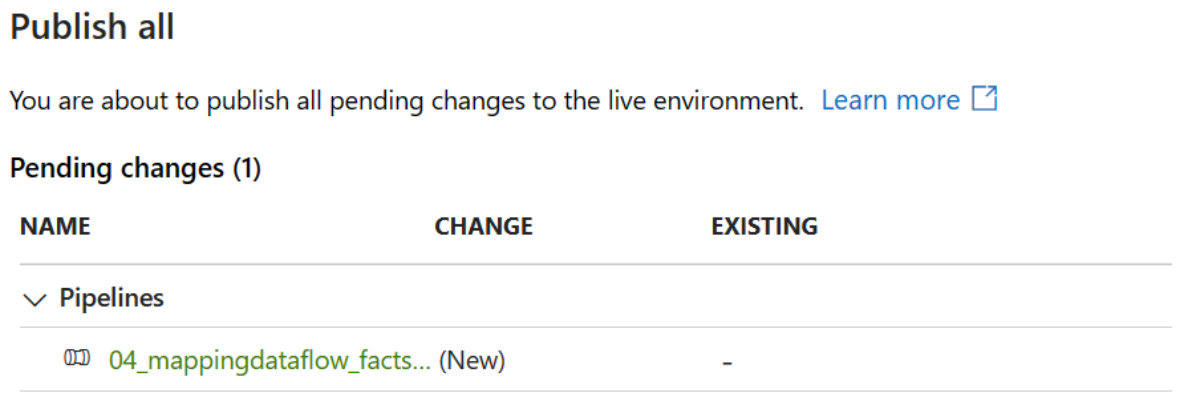
37. After dragging it, give it a name. then move to settings.



38. Then you have to choose staging linked service and choose your service. After that choose the staging storage folder or container which we created earlier.
39. Now if you do not have your staging linked service then you can click on New to create a new service then add the staging container. To link your staging first need to choose your Azure data lake storage gen2 account.

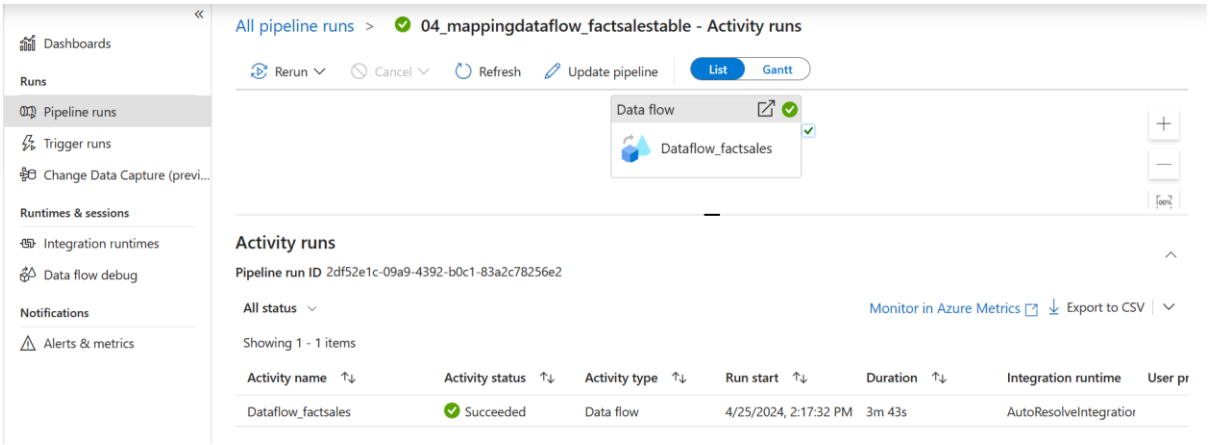


40. After that click on validate all, then click on publish all.



41. Once your publishing is completed click on the trigger to run this pipeline.

42. If you had to use the normal copy data activity using a query, you know the activity would have taken less than 30 seconds. But using a mapping data flow it could take around three to five minutes. And why is the case? It's because, in the background, Azure Data Factory has to start building those Spark clusters that compute infrastructure that will be used for running your mapping data flow. So, it takes time to build up that infrastructure, run your mapping data flow, and then obviously destroy that infrastructure.
43. Here you can see that we are in the monitor section of the pipeline runs. Now you can see that it takes almost 4 minutes to complete.



44. Now go back to SSMS and run the select statement to check whether your data was copied or not.
45. Below you can see that the operation was successful.

SQLQuery1.sql - dat...(sqladminuser (0))*

```
select * from [dbo].[FactSales]
```

150 %

Results | Messages

	ProductID	SalesOrderID	CustomerID	OrderQty	UnitPrice	OrderDate	TaxAmt
1	717	71867	29644	1	858.90	2008-06-01 00:00:00.000	84.7448
2	711	71797	29796	4	20.994	2008-06-01 00:00:00.000	6242.3752
3	780	71902	29929	5	1391.994	2008-06-01 00:00:00.000	5924.7046
4	876	71885	29612	3	72.00	2008-06-01 00:00:00.000	44.0309
5	714	71782	29485	3	29.994	2008-06-01 00:00:00.000	3182.8264
6	936	71846	30102	2	37.254	2008-06-01 00:00:00.000	196.3012
7	891	71796	29660	1	602.346	2008-06-01 00:00:00.000	4610.7707
8	711	71784	29736	2	20.994	2008-06-01 00:00:00.000	8684.9465
9	953	71816	30027	1	728.91	2008-06-01 00:00:00.000	271.8533
10	875	71917	30025	7	5.394	2008-06-01 00:00:00.000	3.2724
11	969	71898	29932	3	1430.442	2008-06-01 00:00:00.000	5118.4791
12	944	71936	30050	3	158.43	2008-06-01 00:00:00.000	7862.2953
13	909	71895	29584	2	23.484	2008-06-01 00:00:00.000	19.7391
14	836	71774	29847	1	356.898	2008-06-01 00:00:00.000	70.4279
15	962	71856	30033	1	445.41	2008-06-01 00:00:00.000	48.1756
16	966	71897	29877	1	1430.442	2008-06-01 00:00:00.000	1014.8712
17	875	71863	29975	11	5.2142	2008-06-01 00:00:00.000	265.9421
18	979	71858	29653	3	445.41	2008-06-01 00:00:00.000	1105.8967
19	869	71831	30019	2	41.994	2008-06-01 00:00:00.000	161.3073
20	738	71783	29957	4	202.332	2008-06-01 00:00:00.000	6708.6741
21	738	71815	30089	1	202.332	2008-06-01 00:00:00.000	91.3263
22	973	71920	29982	1	1020.594	2008-06-01 00:00:00.000	238.4634
23	988	71845	29938	2	112.998	2008-06-01 00:00:00.000	3329.7641
24	973	71938	29546	5	1020.594	2008-06-01 00:00:00.000	7105.029
25	916	71946	29741	1	31.584	2008-06-01 00:00:00.000	3.1163
26	905	71780	30113	4	218.454	2008-06-01 00:00:00.000	3073.4952
27	838	71797	29796	2	858.90	2008-06-01 00:00:00.000	6242.3752
28	981	71902	29929	2	461.694	2008-06-01 00:00:00.000	5924.7046
29	884	71885	29612	6	32.394	2008-06-01 00:00:00.000	44.0309
30	956	71782	29485	3	1430.442	2008-06-01 00:00:00.000	3182.8264
31	739	71846	30102	1	818.70	2008-06-01 00:00:00.000	196.3012