😊 Derived Column

In Azure Data Factory, a derived column is a transformation that allows you to create new columns or modify existing ones based on expressions or calculations applied to the existing data within your dataset. It's a common operation used during data transformation processes to derive new information or modify existing data before loading it into the destination.

Here are some key points about derived columns in Azure Data Factory:

1. **Expression-based Transformation:** Derived columns are created using expressions that can perform various operations such as string manipulation, mathematical calculations, date/time conversions, and conditional logic.
2. **Creation of New Columns:** You can use derived columns to create entirely new columns in your dataset based on the values of existing columns or computed expressions.
3. **Modification of Existing Columns:** Derived columns also allow you to modify the values of existing columns by applying transformations or calculations to them.
4. **Data Cleansing and Enrichment:** Derived columns are often used for data cleansing tasks, such as removing or replacing invalid characters, standardizing formats, or enriching data with additional information.
5. **Flexible and Dynamic:** Derived column transformations offer flexibility in defining expressions, allowing you to dynamically derive or modify columns based on the requirements of your data transformation pipeline.
6. **Integration with Data Flows:** Derived columns can be easily incorporated into Mapping Data Flow activities within Azure Data Factory, providing a visual interface for designing complex data transformation logic without writing code.

Overall, derived columns in Azure Data Factory are powerful tools for manipulating and enriching data as part of your data integration and ETL (Extract, Transform, Load) processes, enabling you to prepare data for analysis, reporting, or loading into downstream systems.

- **In this lab exercise, we are enhancing a fact table, specifically the Fact Sales Table, within Azure Data Factory. Our objective is to add a new column named "Total Amount" to the table, which is calculated by multiplying the "Order Quantity" and "Unit Price" columns. The end goal is to enrich our dataset with this derived information, providing valuable insights for analysis and reporting purposes.**
- **The process begins by dropping the existing Fact Sales Table and recreating it with the additional column in SQL Server Management Studio (SSMS). This sets the stage for our data transformation tasks within Azure Data Factory.**
- **Within Azure Data Factory, we navigate to the existing mapping data flow for the Fact Sales Table. Here, we add a schema modifier to filter the rows of interest and select the desired columns for the final table. Subsequently, we incorporate a derived column transformation to dynamically compute the "Total Amount" based on the provided expression involving the "Order Quantity" and "Unit Price" columns.**

- **Once the derived column is configured, we adjust the sink settings for the Fact Sales Table to reflect the new schema. Mapping adjustments are made to ensure proper handling of the new column.**
- **After ensuring all configurations are in place, we publish the changes to the data flow. We then proceed to execute the associated pipeline, monitoring its run status to confirm successful execution.**
- **Finally, we verify the presence of the new "Total Amount" column in the Fact Sales Table by querying the table in SSMS.**
- **The ultimate aim of this exercise is to empower data analysts and stakeholders with enriched data that includes calculated metrics, facilitating more comprehensive analysis, reporting, and decision-making processes. By adding derived columns such as "Total Amount," we enhance the dataset's informational value, enabling deeper insights into sales performance and trends.**

😄 To begin with the Lab:

1. Now in this lab we are going to add a new column to our Fact Table.
2. Below, if you see in our fact sales table, we are going to add a new column with the name Total amount which will be created by multiplying order quantity and unit price.

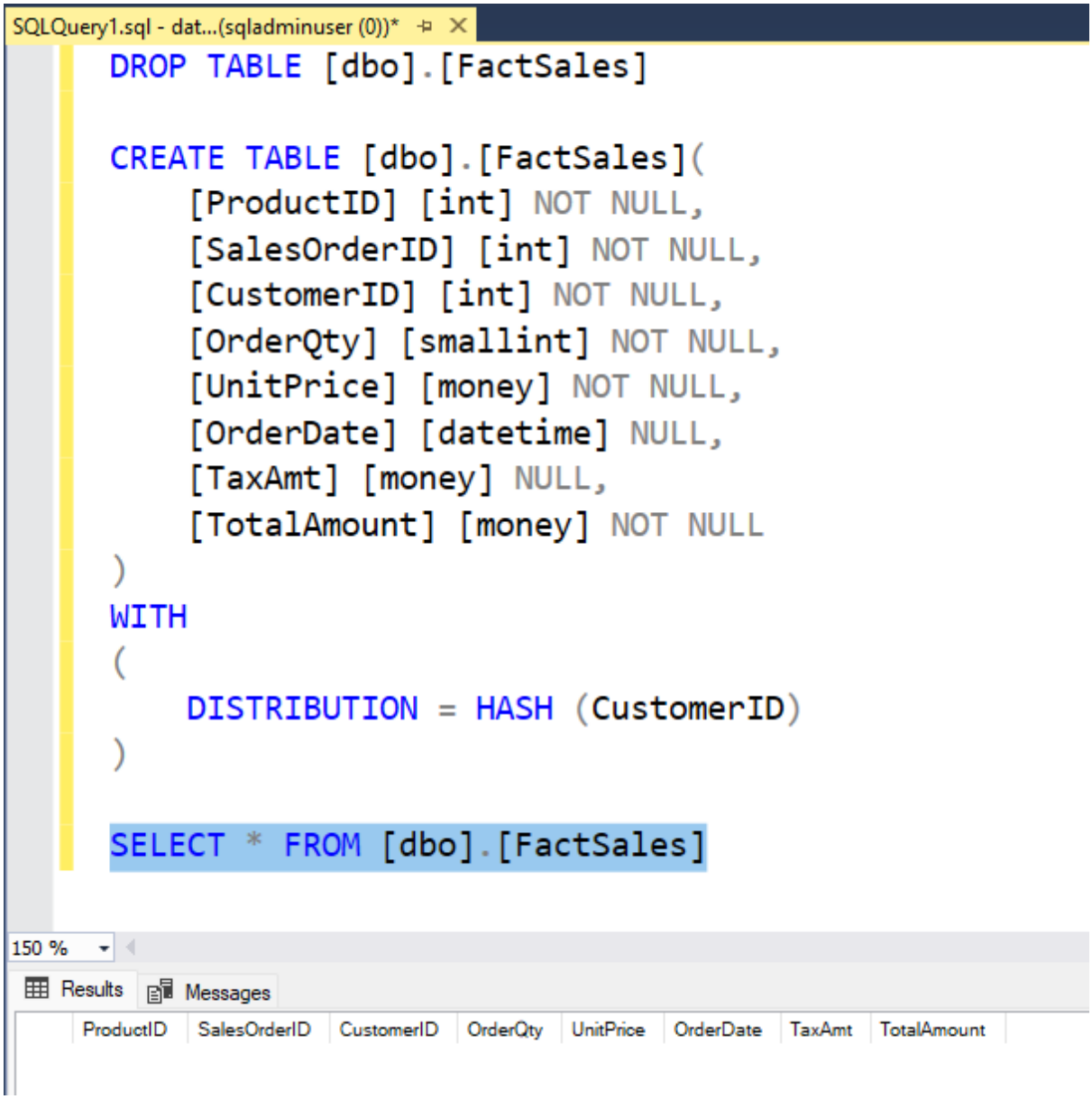
```
SELECT * FROM FactSales
```

150 %

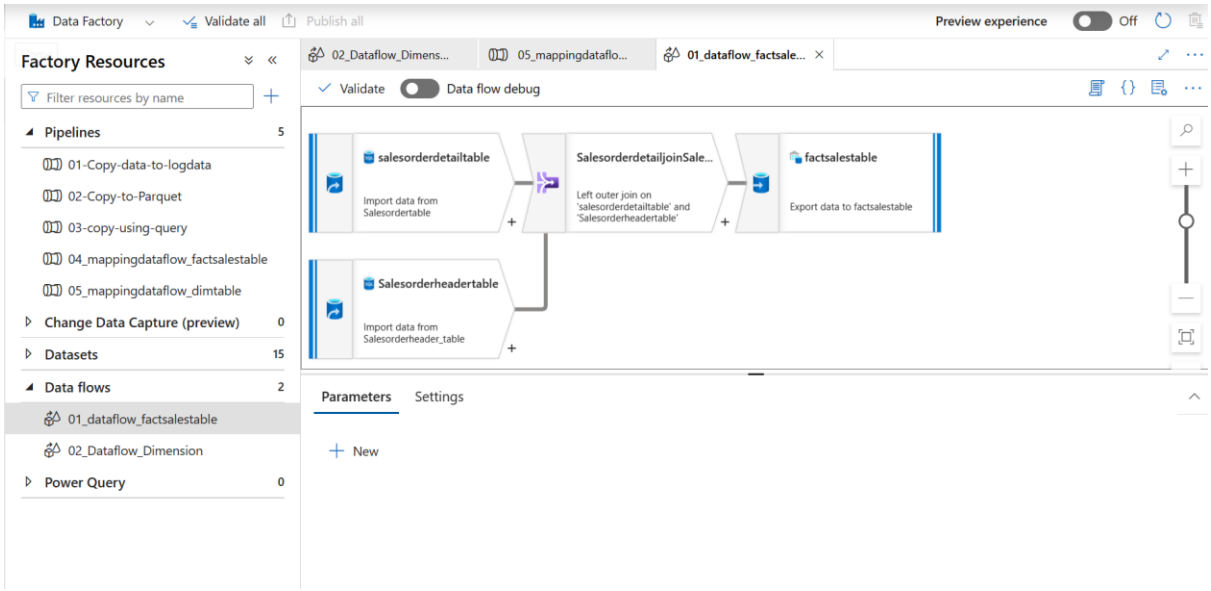
Results | Messages

	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

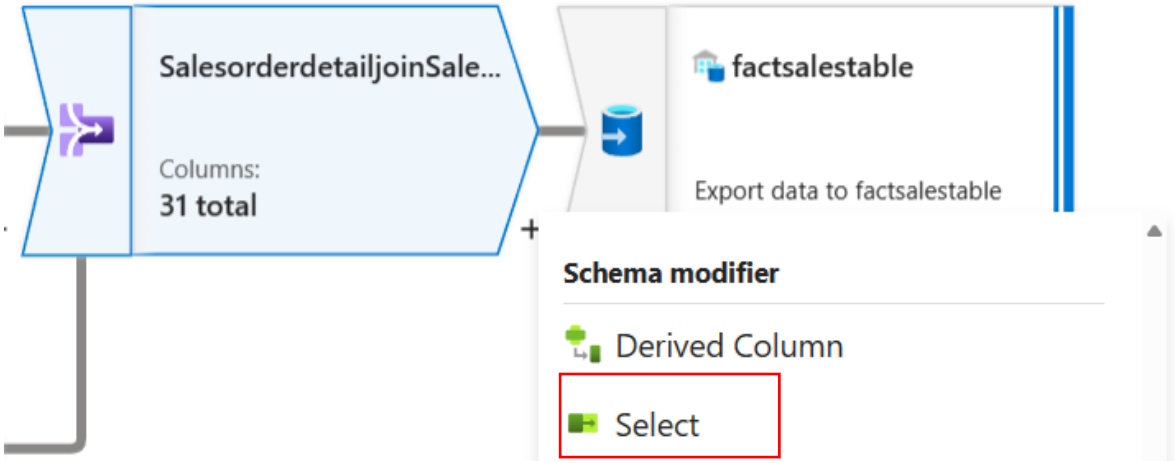
3. But for that first we are going to drop the fact sales table and recreate a new table with the extra column in it.
4. Below you can see that first, we dropped our fact sales table. Then we recreated it with that extra column. And if you run the select statement, you will see that currently our table is empty.



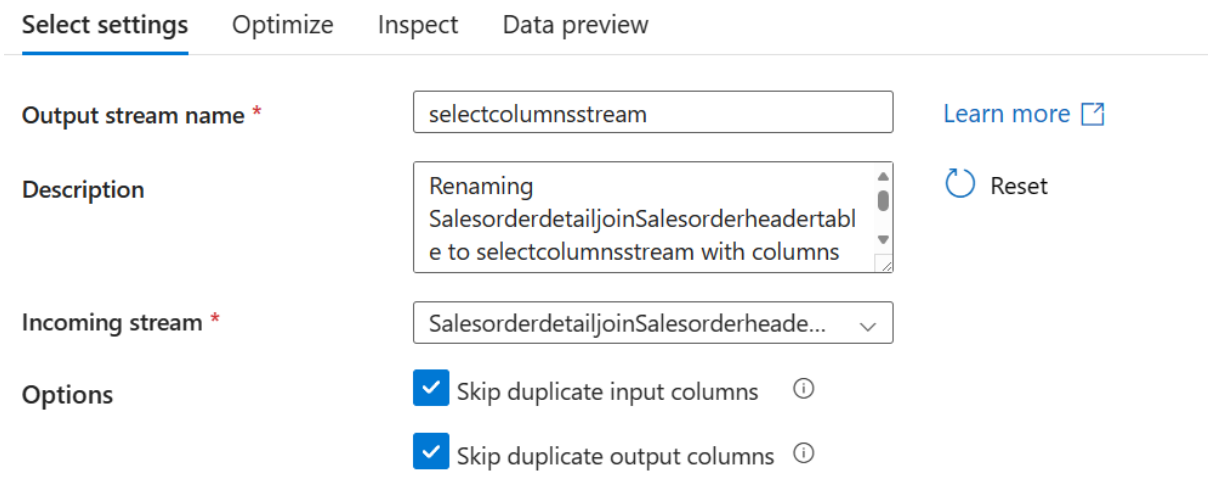
5. Now we need to navigate to the Data Factory wizard.
6. And we need to go to our existing mapping data flow for the fact sales table.
7. Below you can see that we are on the data flow for our fact sales table.



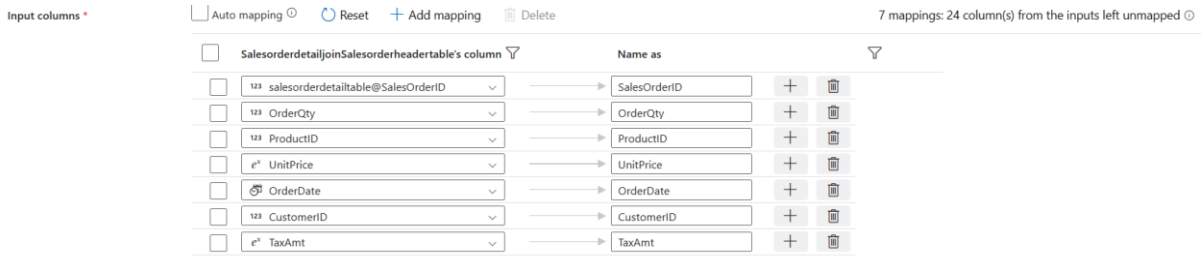
8. Here you need to select the join then click on the plus icon to add the schema modifier of Select to only the rows of interest to us of our entire stream.



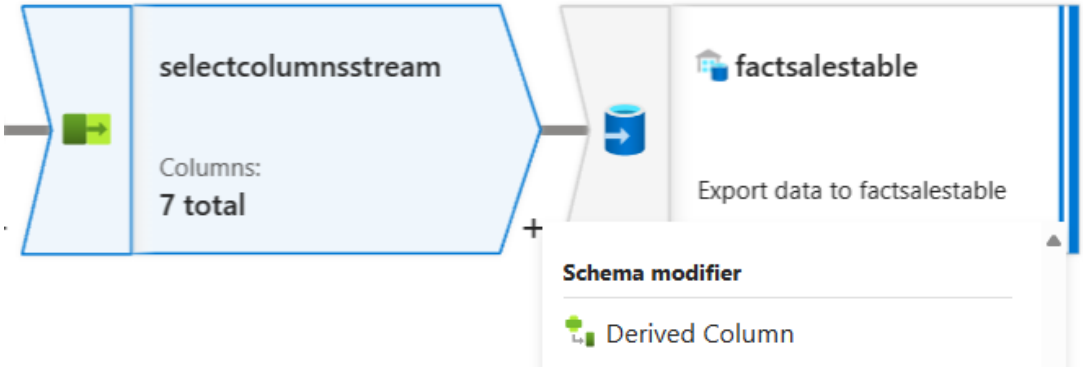
9. From here we will select the column of our interest that will go to the final fact sales table.
10. Then in the select settings first you need to give it a name then scroll down.



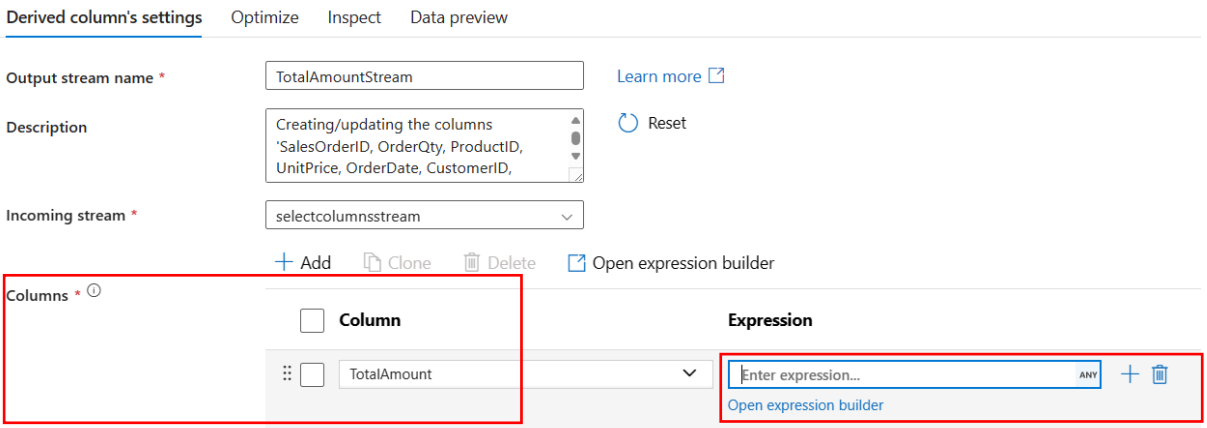
11. Then in the input columns disable auto mapping and you will see that we had multiple mapping. So, you need to remove the extra mapping and keep only what we desire as shown below.
12. You need to choose the mappings shown below only. Delete the other mappings.



13. After that from your select column stream you need to click on plus icon then from the schema modifier this time you need to choose derived column.

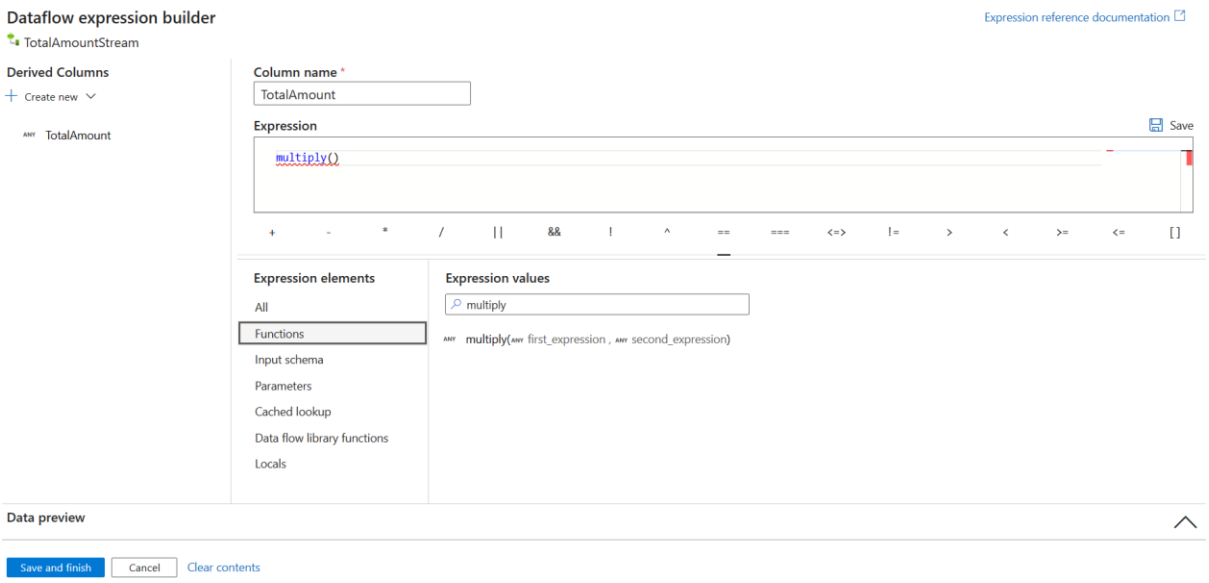


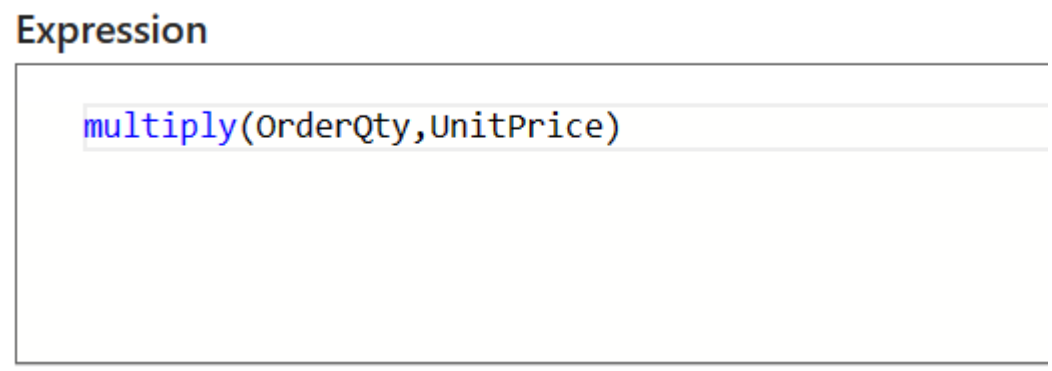
14. Now in the derived column settings, first name it. Then in the column you need to write the column name, after that click on the expression, there you will see an option to open expression builder. Open it.

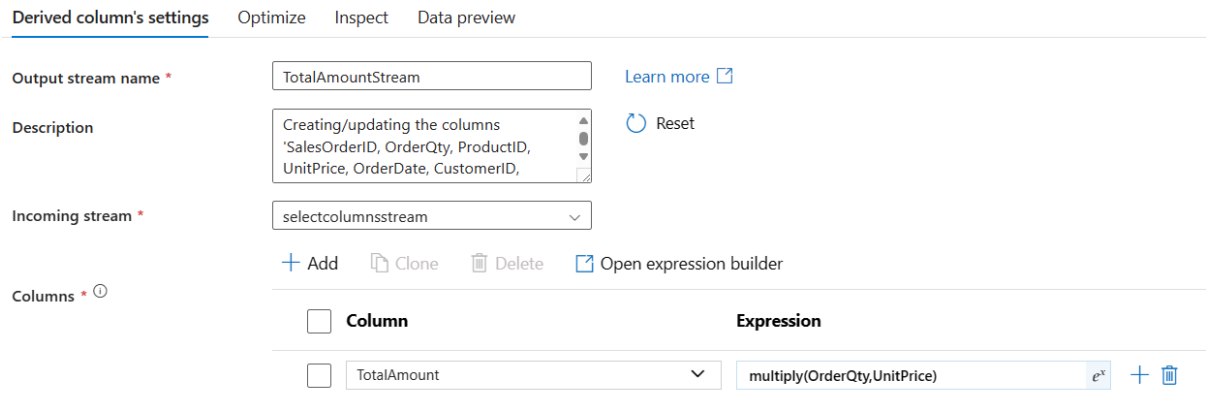


15. The expression builder can now be used to create an expression that can dynamically create the value that you want to store in this column.
16. Here we want to multiply the order quantity and unit price.
17. First you have to select functions then search multiply then select the option for it.

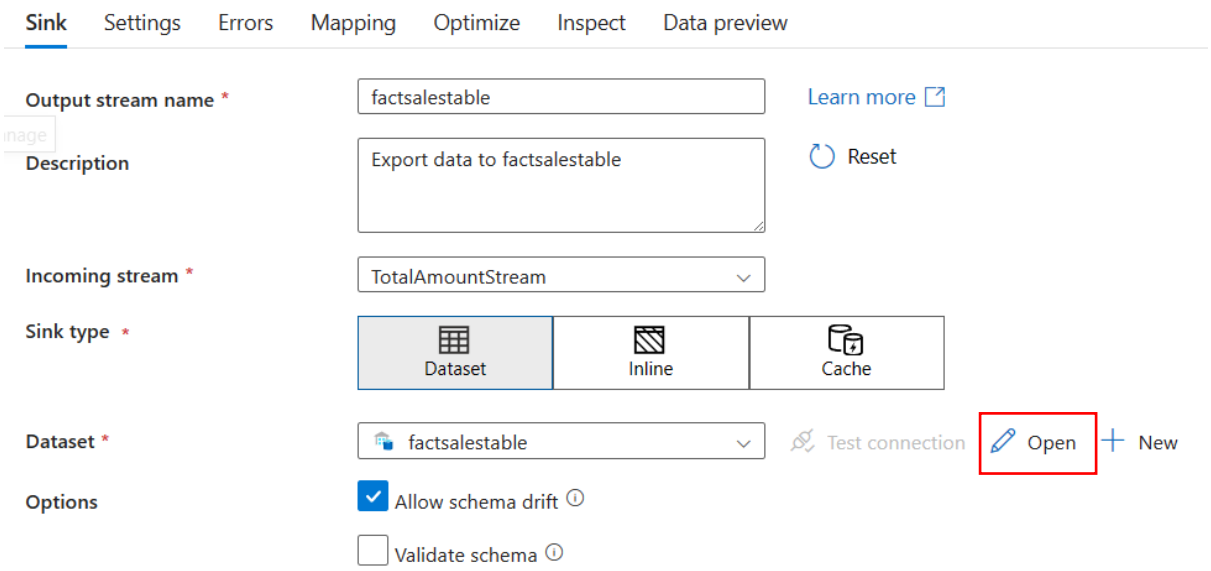
18. After that select input schema and from there choose order quantity and unit price. After that click on save and finish.







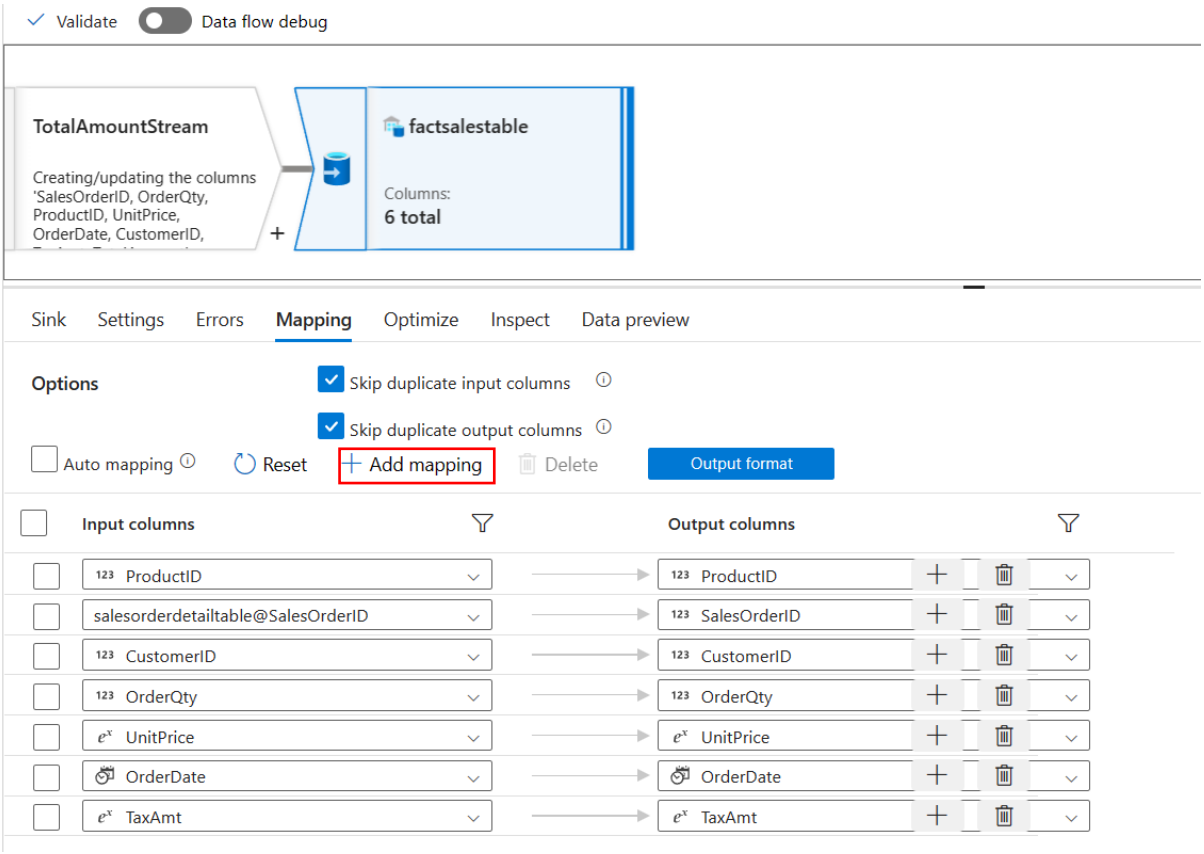
19. Now we need to go to our fact sales table mapping which is also our sink.
20. In sink we need to click on open.



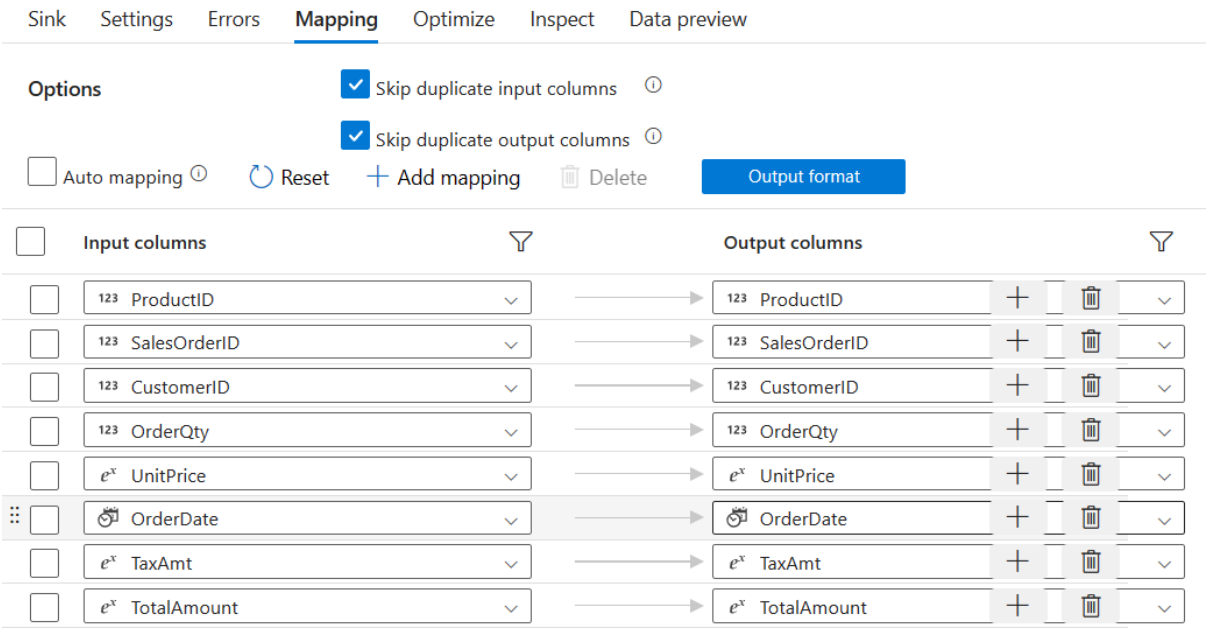
21. Then we need to go to schema, there click on import schema and you will see your new column in place.

Column name	Type
ProductID	int
SalesOrderID	int
CustomerID	int
OrderQty	smallint
UnitPrice	money
OrderDate	datetime
TaxAmt	money
TotalAmount	money

22. Here we need to go to mappings and hit reset.
23. Then we need to click on add mapping choose fixed mapping and add our new column in it which is the total amount.



24. Below you can see that we have added our new column.



25. Now that we have everything in place click on publish all. Below you can see that what it will going to publish.

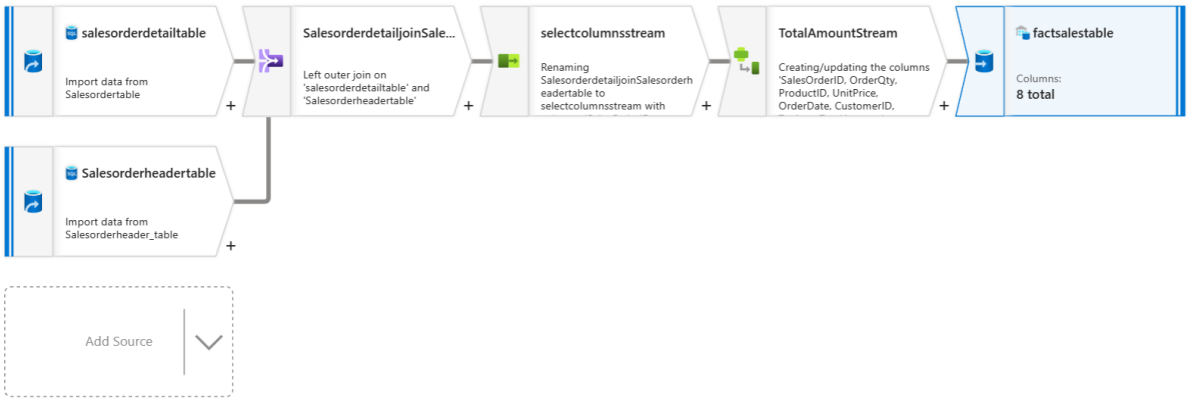
Publish all

You are about to publish all pending changes to the live environment. Learn more

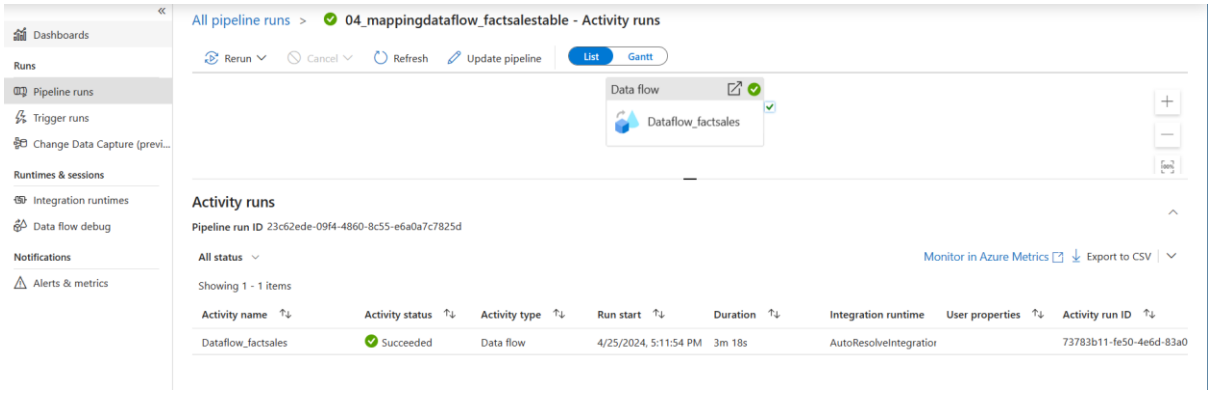
Pending changes (2)

NAME	CHANGE	EXISTING
Datasets		
factsalestable	(Edited)	factsalestable
Data flows		
01_dataflow_factsalestable	(Edited)	01_dataflow_factsalestable

26. Here you can see that data flow mapping diagram.



27. Now the last step is to go to the pipeline for fact sales and just trigger it to run the pipeline again.
28. From the monitor section you can see that the pipeline run was successful.



29. Now go to SSMS and run the select statement to check for your data.
30. Below you can see that new column in place.

```
SELECT * FROM [dbo].[FactSales]
```

150 %

Results | Messages

	ProductID	SalesOrderID	CustomerID	OrderQty	UnitPrice	OrderDate	TaxAmt	TotalAmount
1	717	71867	29644	1	858.90	2008-06-01 00:00:00.000	84.7448	858.90
2	711	71797	29796	4	20.994	2008-06-01 00:00:00.000	6242.3752	83.976
3	881	71899	29568	2	32.394	2008-06-01 00:00:00.000	193.2538	64.788
4	876	71885	29612	3	72.00	2008-06-01 00:00:00.000	44.0309	216.00
5	714	71782	29485	3	29.994	2008-06-01 00:00:00.000	3182.8264	89.982
6	936	71846	30102	2	37.254	2008-06-01 00:00:00.000	196.3012	74.508
7	870	71923	29781	1	2.994	2008-06-01 00:00:00.000	8.5233	2.994
8	711	71784	29736	2	20.994	2008-06-01 00:00:00.000	8684.9465	41.988
9	953	71816	30027	1	728.91	2008-06-01 00:00:00.000	271.8533	728.91
10	907	71776	30072	1	63.90	2008-06-01 00:00:00.000	6.3048	63.90
11	969	71898	29932	3	1430.442	2008-06-01 00:00:00.000	5118.4791	4291.326
12	944	71936	30050	3	158.43	2008-06-01 00:00:00.000	7862.2953	475.29
13	909	71895	29584	2	23.484	2008-06-01 00:00:00.000	19.7391	46.968
14	836	71774	29847	1	356.898	2008-06-01 00:00:00.000	70.4279	356.898
15	962	71856	30033	1	445.41	2008-06-01 00:00:00.000	48.1756	445.41
16	966	71897	29877	1	1430.442	2008-06-01 00:00:00.000	1014.8712	1430.442
17	875	71863	29975	11	5.2142	2008-06-01 00:00:00.000	265.9421	57.3562
18	979	71858	29653	3	445.41	2008-06-01 00:00:00.000	1105.8967	1336.23
19	869	71831	30019	2	41.994	2008-06-01 00:00:00.000	161.3073	83.988
20	738	71783	29957	4	202.332	2008-06-01 00:00:00.000	6708.6741	809.328
21	738	71815	30089	1	202.332	2008-06-01 00:00:00.000	91.3263	202.332
22	836	71915	29638	3	356.898	2008-06-01 00:00:00.000	170.9785	1070.694
23	988	71845	29938	2	112.998	2008-06-01 00:00:00.000	3329.7641	225.996

Query executed successfully.