Transfer data from Azure Blob Storage (Gen2) to SQL database

The tutorial I will explain how to transfer data from Azure Blob Storage (Gen2) to a SQL database using Data Factory. It involves creating data sets for the CSV file and SQL table, configuring the copy activity, and verifying the successful transfer. Debugging the schema mapping ensures the file matches the table.

Initially, the schema and table are created in Vscode, followed by creating a new folder and pipeline in the Data Factory. New tables are created for the source (CSV file) and the sink (SQL database table).

For the CSV file, the linked service is set as the **Blob Storage**, specifying the file path and delimiter. The Data Factory automatically reads columns due to the file having a header, serving as instructions for reading the data.

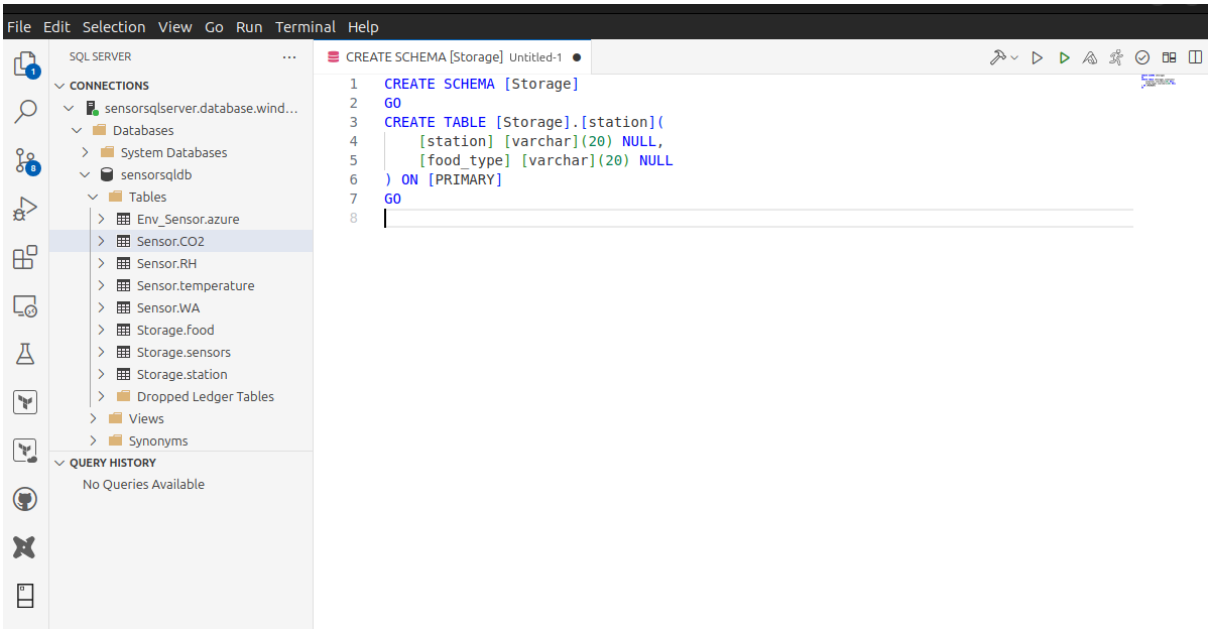
The database table is defined as a dataset in Data Factory, with the linked service set as the **SQL Database** and the table name provided. Data Factory reads columns from the actual table, used as instructions for writing the data.

The copy activity is then configured in the pipeline, selecting the source (CSV file) and destination (SQL table). The **Mapping** tab visualises how the source data will be transformed and loaded into the destination database. Data Factory automatically matches columns from the CSV file to the corresponding columns in the database table, however, if the schema does not match you can manually edit the JSON file for schema matching.

After configuration, the pipeline can be validated and debugged to copy the CSV file to the SQL database. The amount of data transferred can be checked by comparing the rows read and written. Finally, the data from the SQL database can be utilised.

Steps:

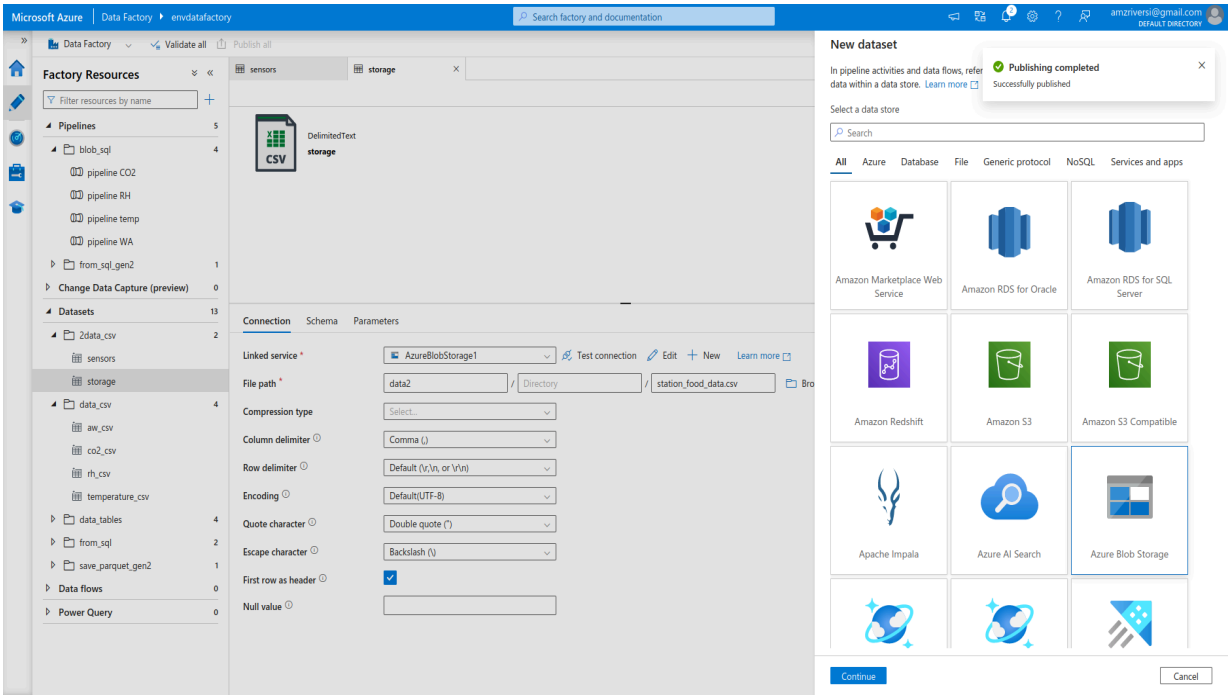
1. First, we need to create the schema and the tables. To execute this you can connect to Azure Server in Vscode and remotely create a database.



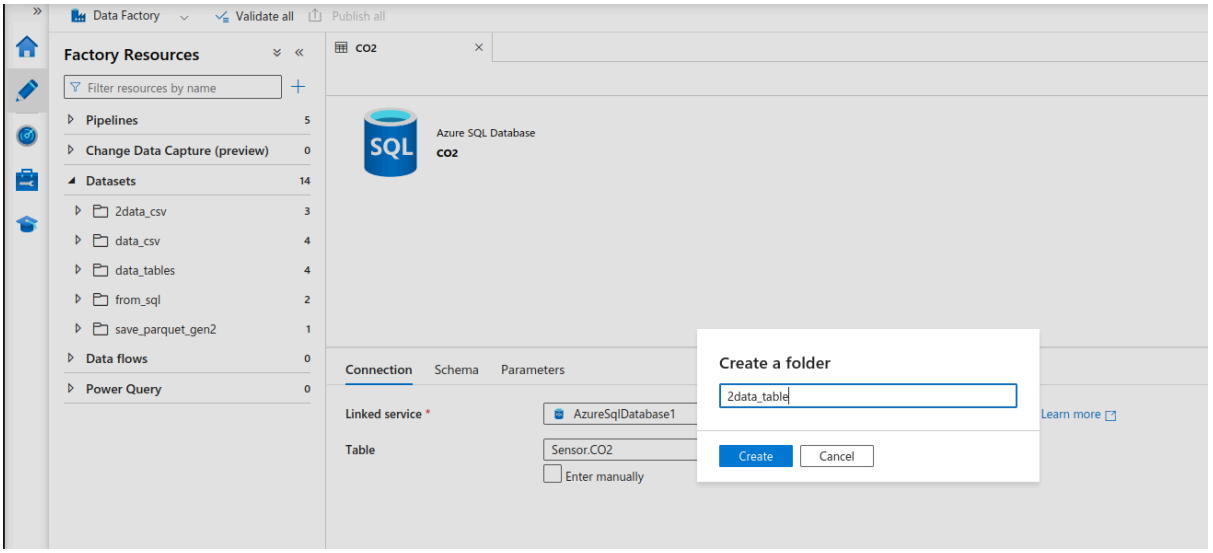
2. In Data Factory

a. Create a new folder and then a new pipeline within that folder. Next, create new folders, one for tables that will be your source (csv) and other for tables that will be the sink (destination).

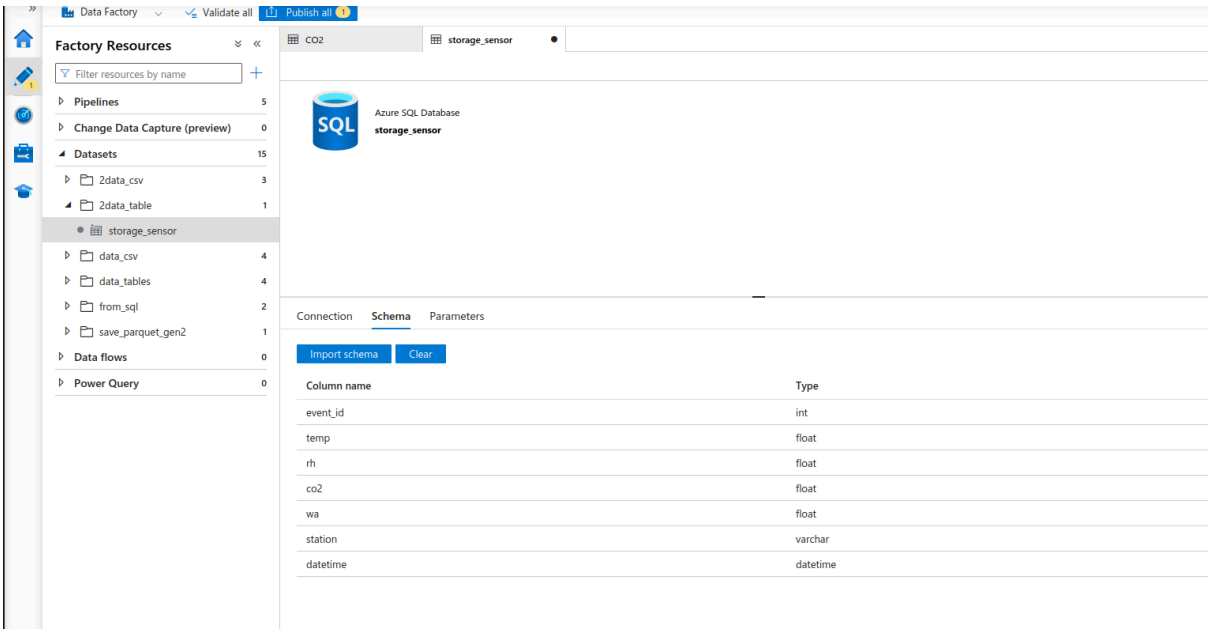
To begin, specify the linked service for the CSV file, which in this case is the **Blob Storage**. Provide the file path and specify the delimiter as a comma. Since the CSV file has a header, Data Factory can automatically read the columns from the file. This data set will serve as instructions for Data Factory to read the data.



b. Next, define the database table as a dataset in Data Factory. Specify the linked service as the SQL database and provide the table name.

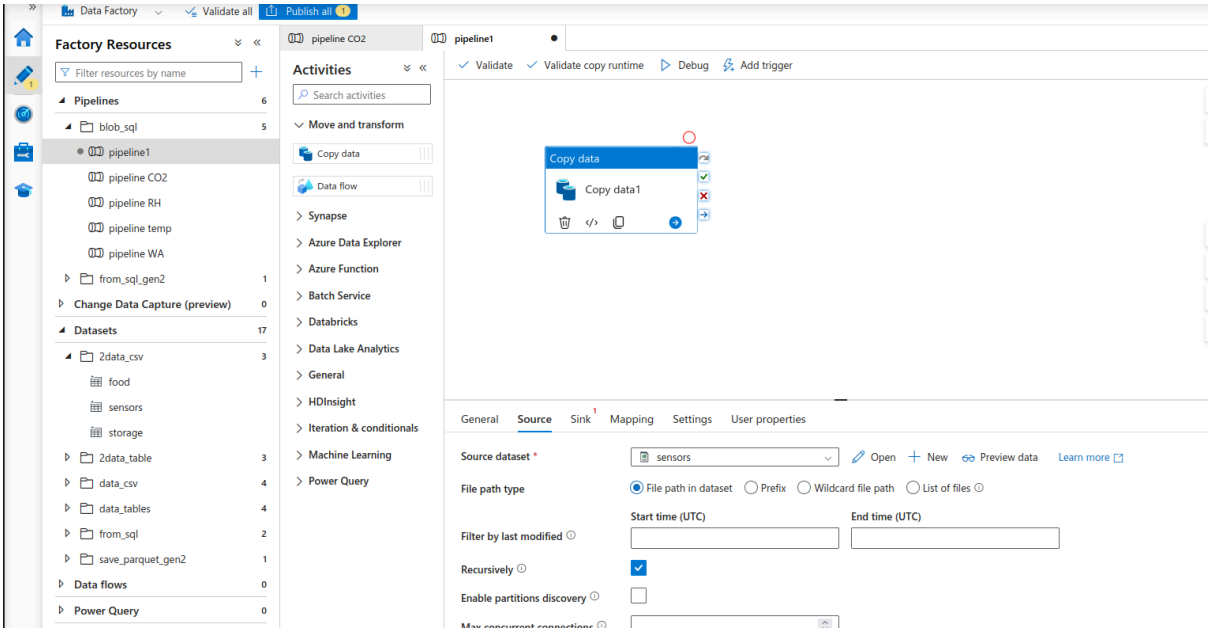


c. Once specified the table name, Data Factory can read the columns from the actual table. These columns will be used as instructions for Data Factory to write the data to the database.

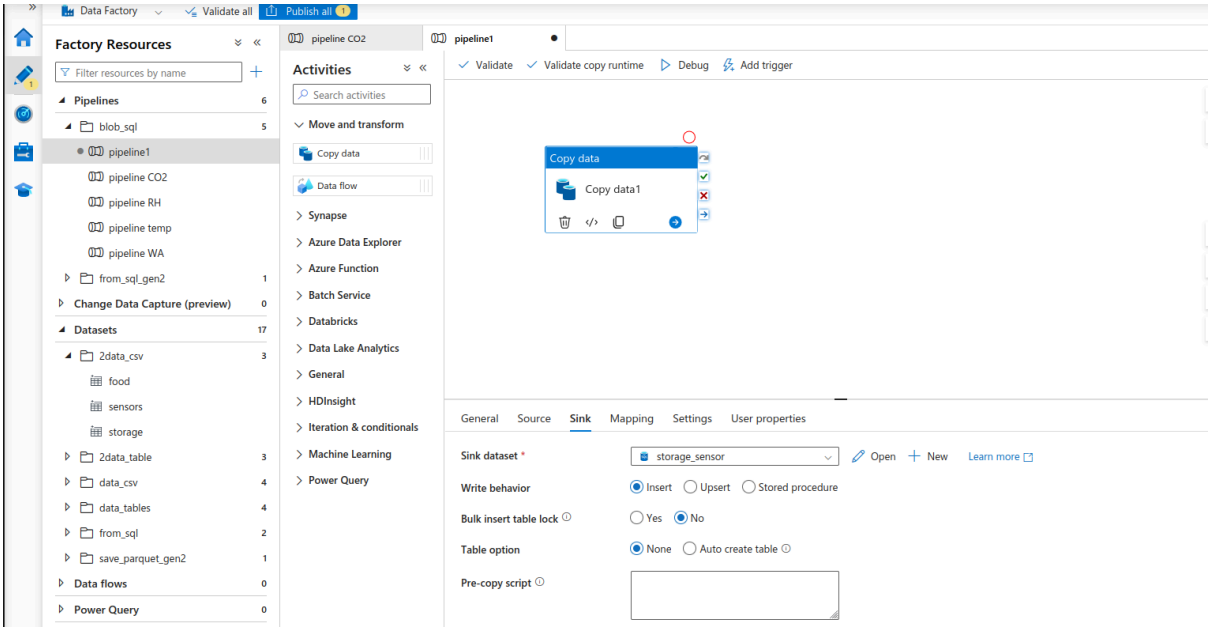


After completing these steps, configure the copy activity in the pipeline, add the **Copy file** that you can find in the pipeline **Activities tab,** and let Data Factory handle the copying process.

- First select the **source** (csv files)

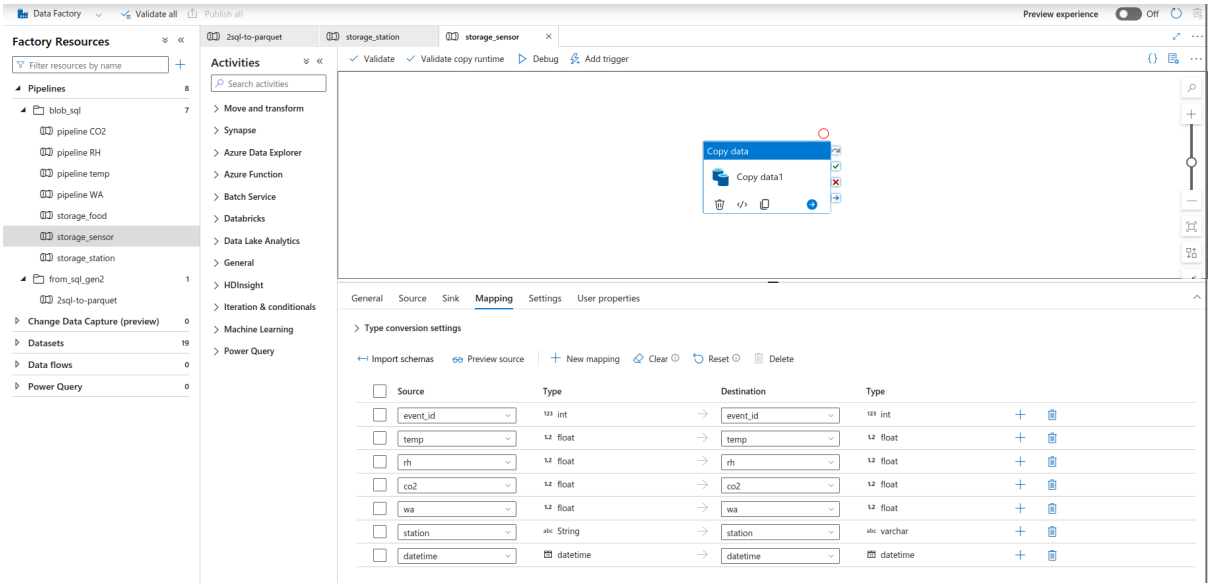


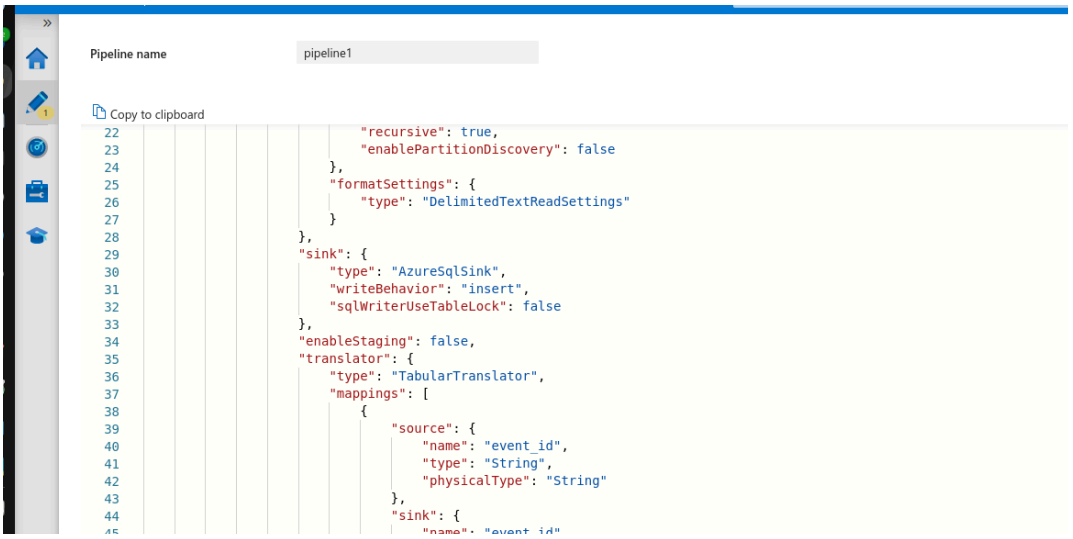
d. Select destination (**sink**) for the file, in this example I have added all tables that will be the destination for the csv files in a folder 2data_table.
Select the table and do **Mapping.**



e. The mapping tab allows us to easily see how our source data will be transformed and loaded into our destination database. By importing the schemas, Data Factory can automatically match up the columns from our CSV file to the corresponding columns in our database table. This seamless mapping process ensures that our data is accurately transferred without any manual intervention required.

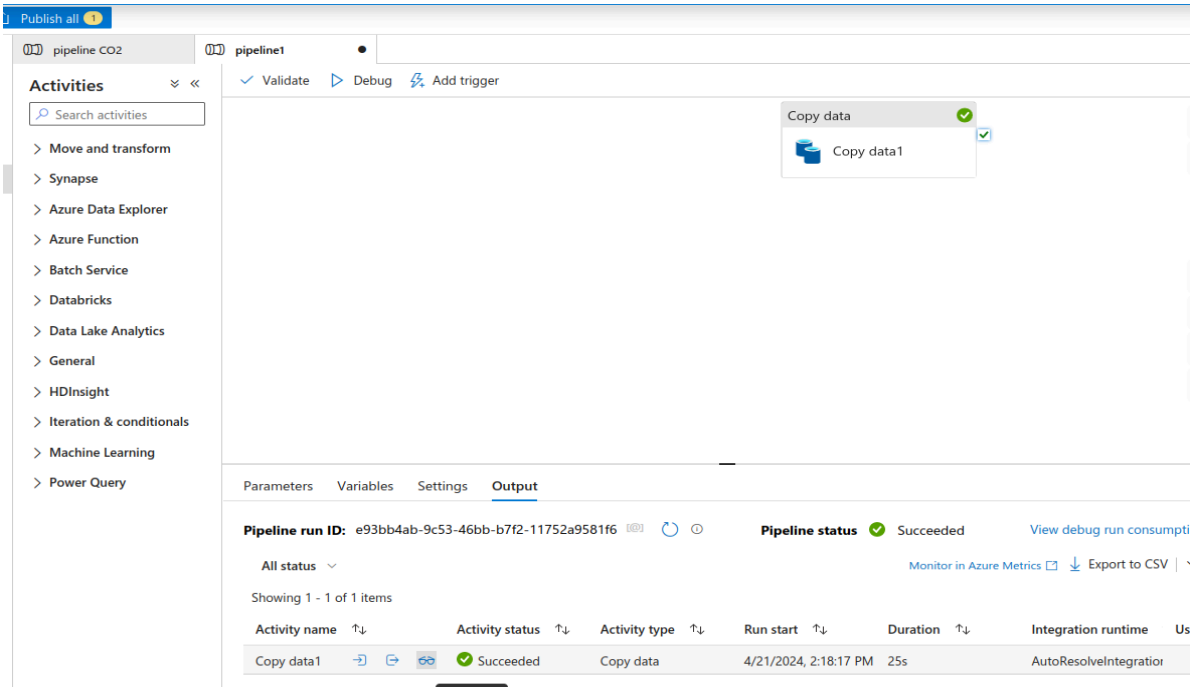
However, if the schema does not match you can manually edit the json file (you can find the json file by clicking on the icon **{ }**) →



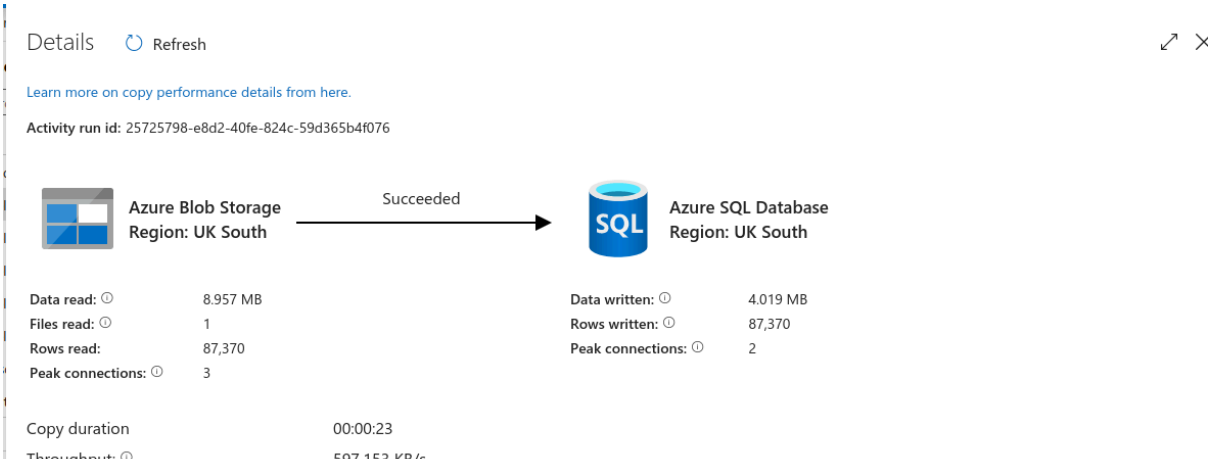


4. Copy to SQL

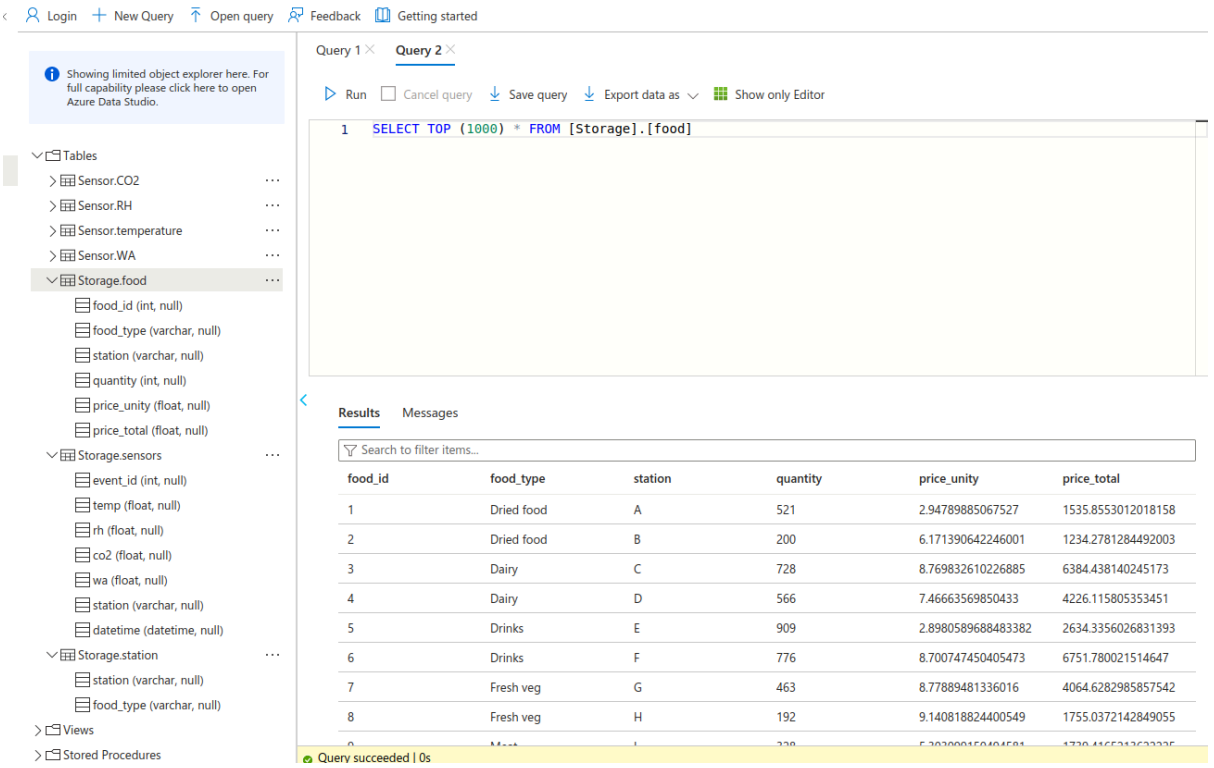
After you can copy the csv file to the database:
Validate the pipeline, if everything is ok you can **Debug** and the data will be copied to SQL database.



If succeeded you can check the amount of data transferred (rows read - rows written)



Now you can use data from SQL databases!



Summary

Here I have explained the process of transferring data from Azure Blob Storage to a SQL database using Data Factory, including steps such as creating data sets, configuring the copy activity, and debugging the schema mapping. The was successfully transferred and read for further analysis or processing.