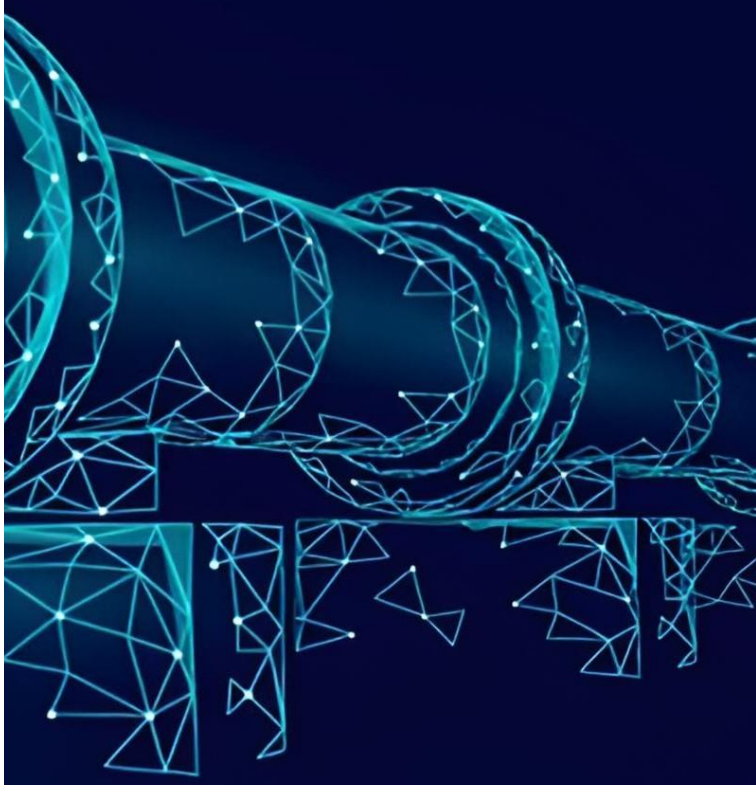
Project: Data Integration Pipelines for NYC Payroll Data Analytics



Jul 17 2024

Dahi Nemutlu

Contents

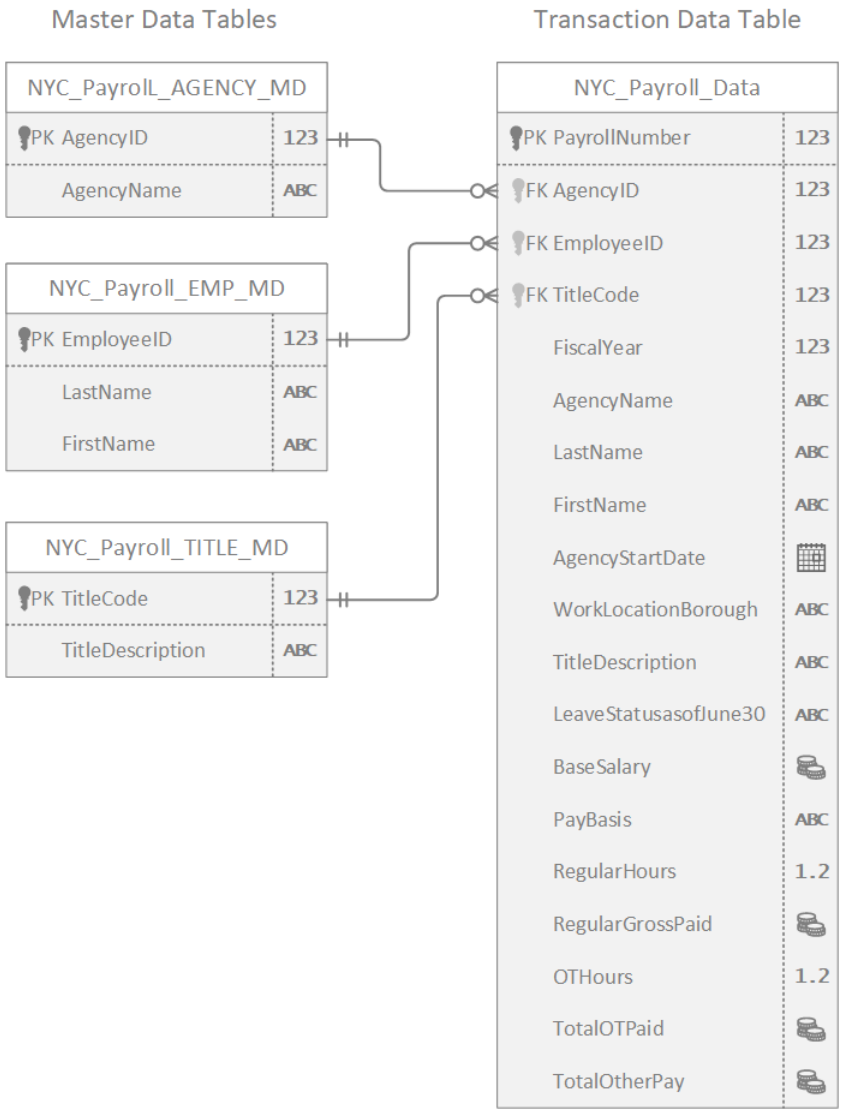
1. Project Overview

The City of New York would like to develop a Data Analytics platform on Azure Synapse Analytics to accomplish two primary objectives:

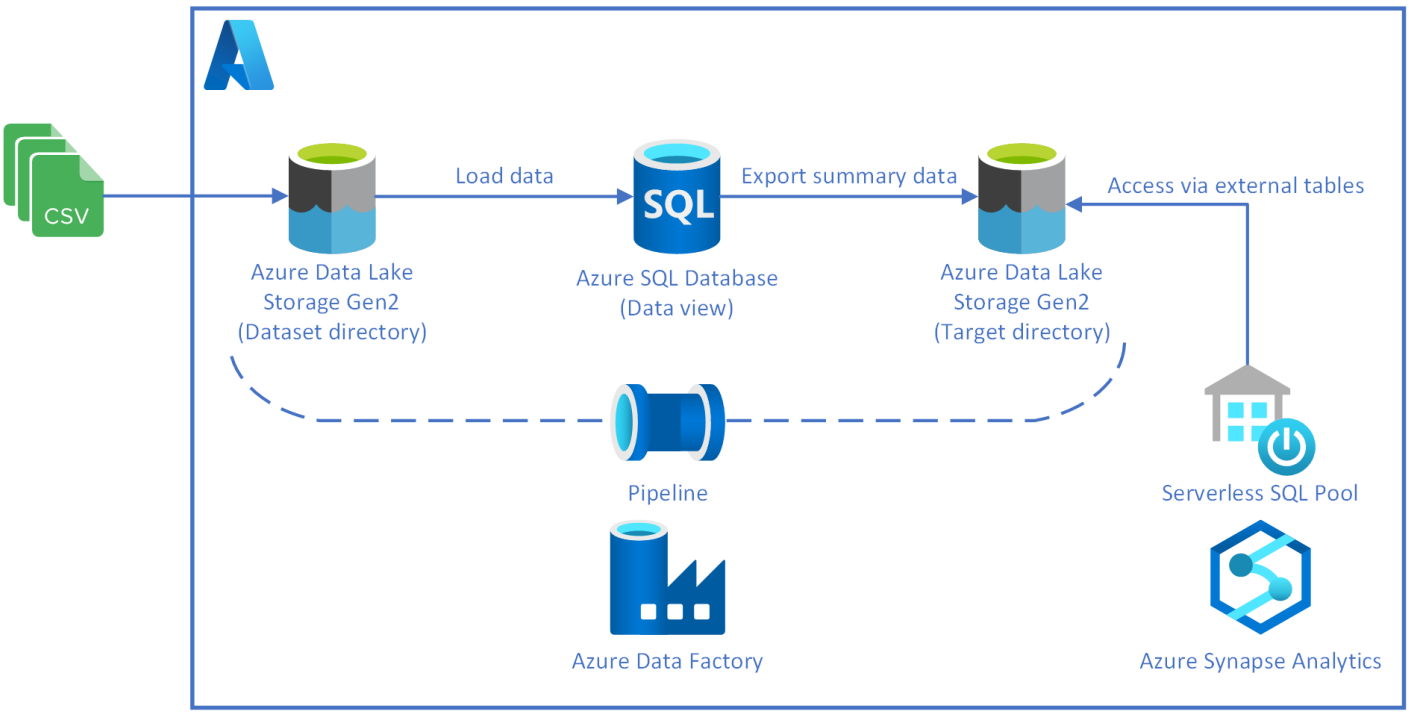
1. Analyze how the City's financial resources are allocated and how much of the City's budget is being devoted to overtime.
2. Make the data available to the interested public to show how the City's budget is being spent on salary and overtime pay for all municipal employees.

With this project, we aim to create high-quality data pipelines that are dynamic, automated, and monitored for efficient operation.

The source data resides in Azure Data Lake and needs to be processed in a NYC data warehouse. The source datasets consist of CSV files with Employee master data and monthly payroll data entered by various City agencies.



In the following pages, we will go through the project instructions and by the end we will have built a Data Integration Pipelines on the NYC Payroll Data. We will be using Azure Data Factory to create Data views in Azure SQL DB from the source data files in DataLake Gen2. Then we build our dataflows and pipelines to create payroll aggregated data that will be exported to a target directory in DataLake Gen2 storage over which Synapse Analytics external table is built. At a high level the pipeline will look like below:



2. Create and Configure Resources

For this project, we'll do our work in the Azure Portal, using several Azure resources including:

- Azure Data Lake Gen2
- Azure SQL DB
- Azure Data Factory
- Azure Synapse Analytics

2.1. Create the Data Lake and Upload Data

Create an Azure Data Lake Storage Gen2 (storage account) and associated storage container resource named **adlsnycpayroll-yourfirstname-lastintial**.

In the Azure Data Lake Gen2 creation flow, go to Advanced tab and ensure below options are checked:

- Require secure transfer for REST API operations
- Allow enabling anonymous access on individual containers
- Enable storage account key access
- Default to Microsoft Entra authorization in the Azure portal
- Enable hierarchical namespace

Create two directories in this storage container named:

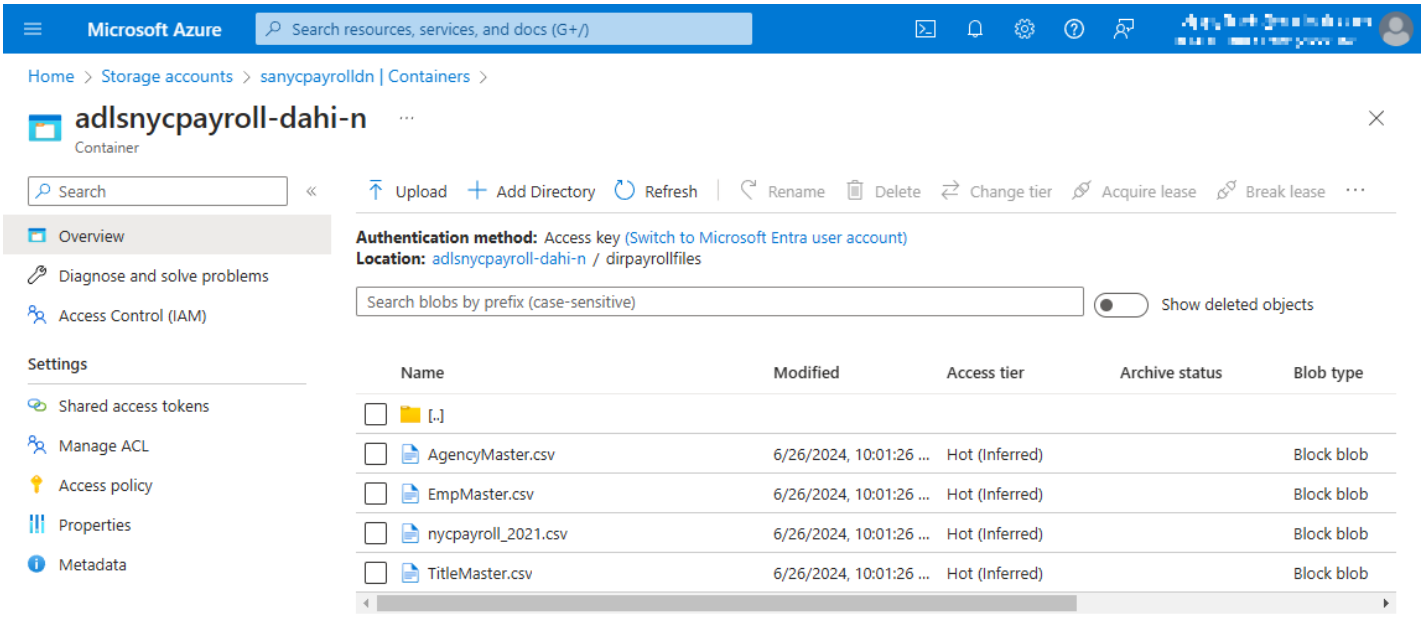
- dirpayrollfiles
- dirhistoryfiles

Upload these files from the project data to the **dirpayrollfiles** folder:

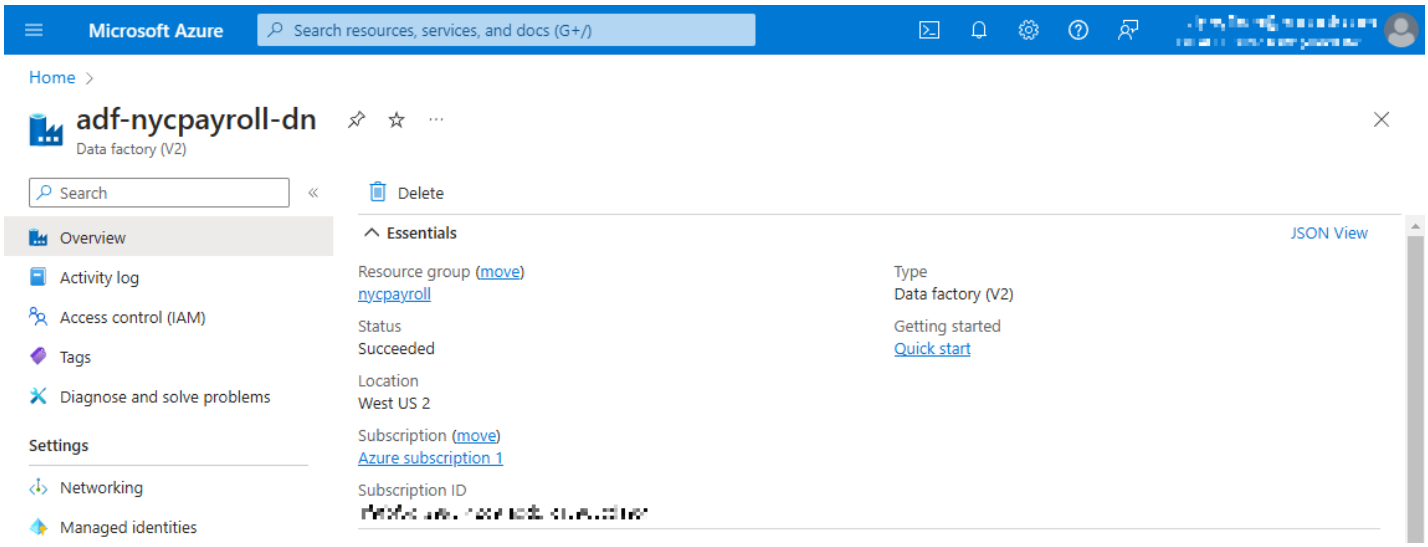
- EmpMaster.csv
- AgencyMaster.csv
- TitleMaster.csv
- nycpayroll_2021.csv

Upload this file (historical data) from the project data to the **dirhistoryfiles** folder:

- nycpayroll_2020.csv



2.2. Create an Azure Data Factory Resource



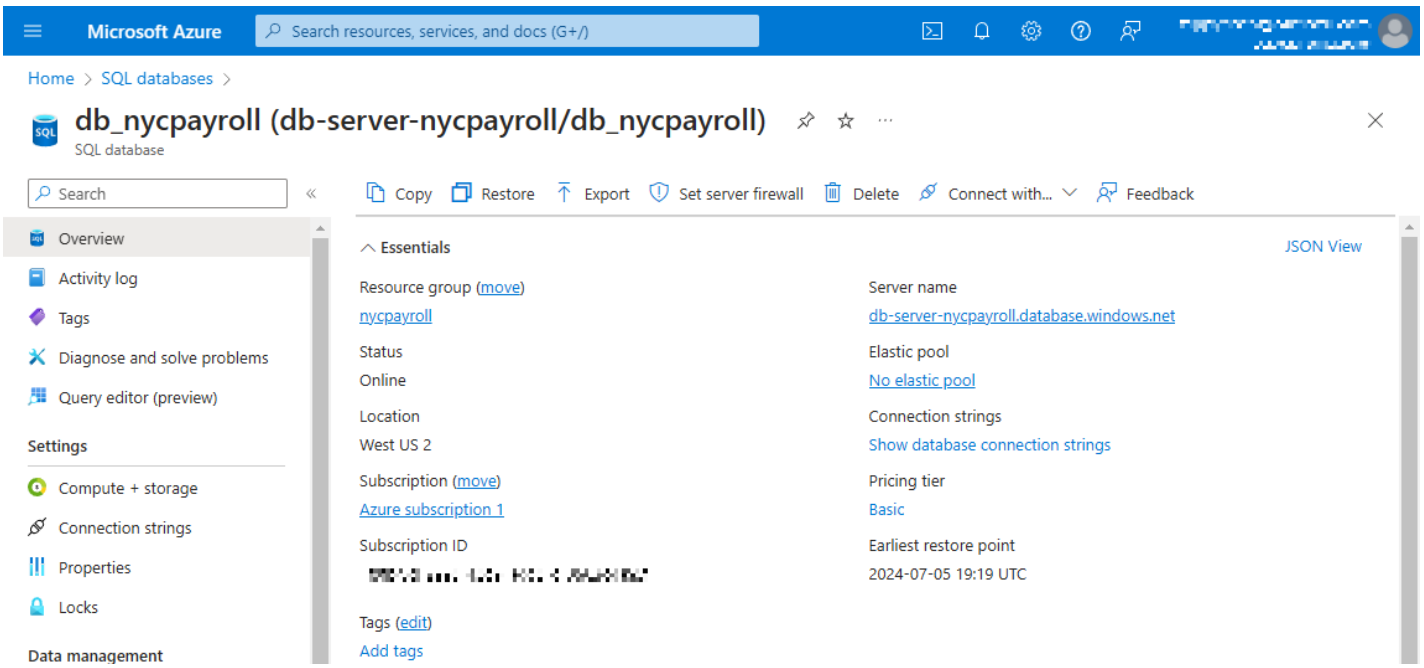
2.3. Create a SQL Database

In the Azure portal, create a SQL Database resource named **db_nycpayroll**.

In the creation steps, you will be required to create an SQL server, create a server with Service tier: Basic (For less demanding workloads).

In Networking tab, allow both below options:

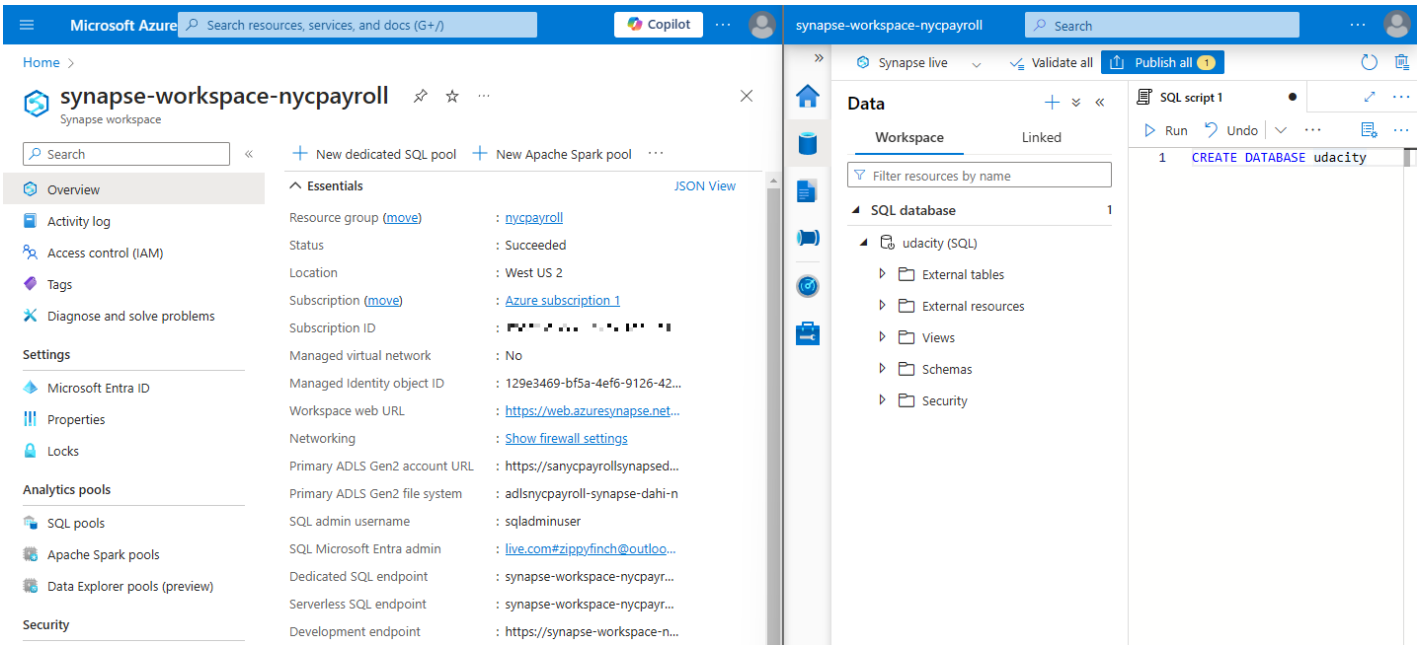
- Allow Azure services and resources to access this server
- Add current client IP address



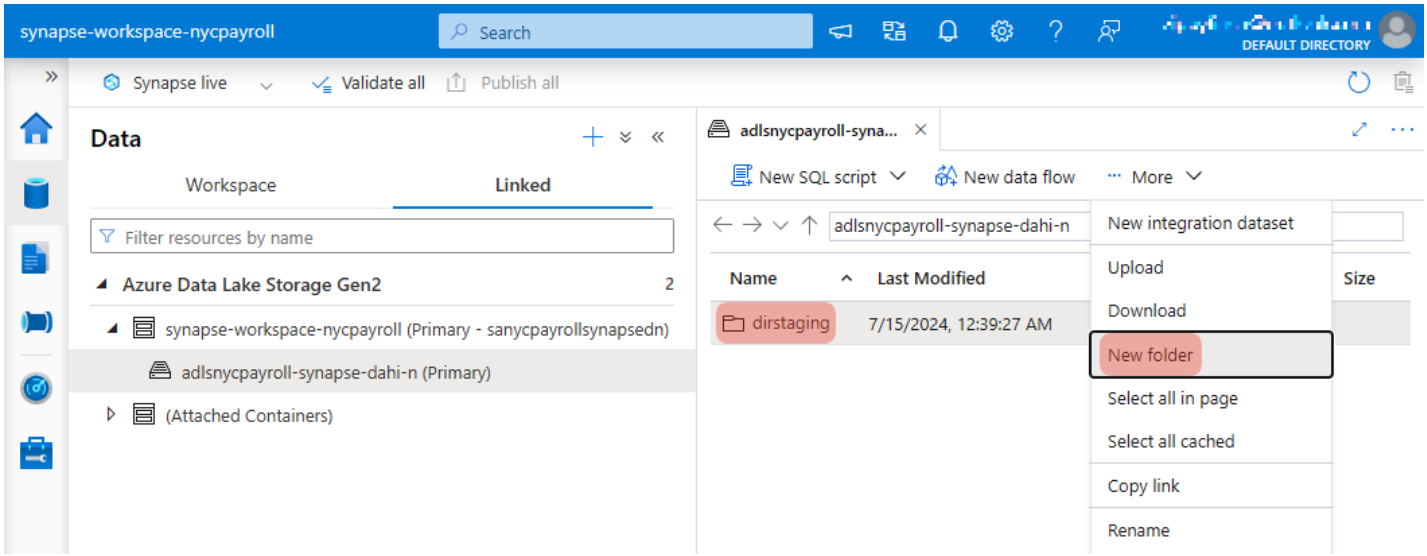
2.4. Create a Synapse Analytics Workspace

- You are only allowed one Synapse Analytics workspace per Azure account, a Microsoft restriction.
- Create a new Azure Data Lake Gen2 and file system for Synapse Analytics when you are creating the Synapse Analytics workspace in the Azure portal.
- Under Synapse, you will not be allowed to run SQL commands in the default main database. Use the below command to create a database and then refresh the database selector dropdown and choose your created database before running any queries.

```
CREATE DATABASE udacity
```



- Create a folder named **dirstaging** in this storage, which will be used by the pipelines we will create as part of the project to store staging data for integration with Azure Synapse. This will be discussed in further pages.



2.5. Create External Table for Summary Data in Synapse Analytics

Define the file format, if not already, for reading/saving the data from/to a comma delimited file in blob storage.

```
IF NOT EXISTS (
    SELECT *
    FROM sys.external_file_formats
    WHERE name = 'SynapseDelimitedTextFormat'
) CREATE EXTERNAL FILE FORMAT [SynapseDelimitedTextFormat] WITH (
    FORMAT_TYPE = DELIMITEDTEXT,
    FORMAT_OPTIONS (
        FIELD_TERMINATOR = ',',
        USE_TYPE_DEFAULT = FALSE
    )
)
GO
```

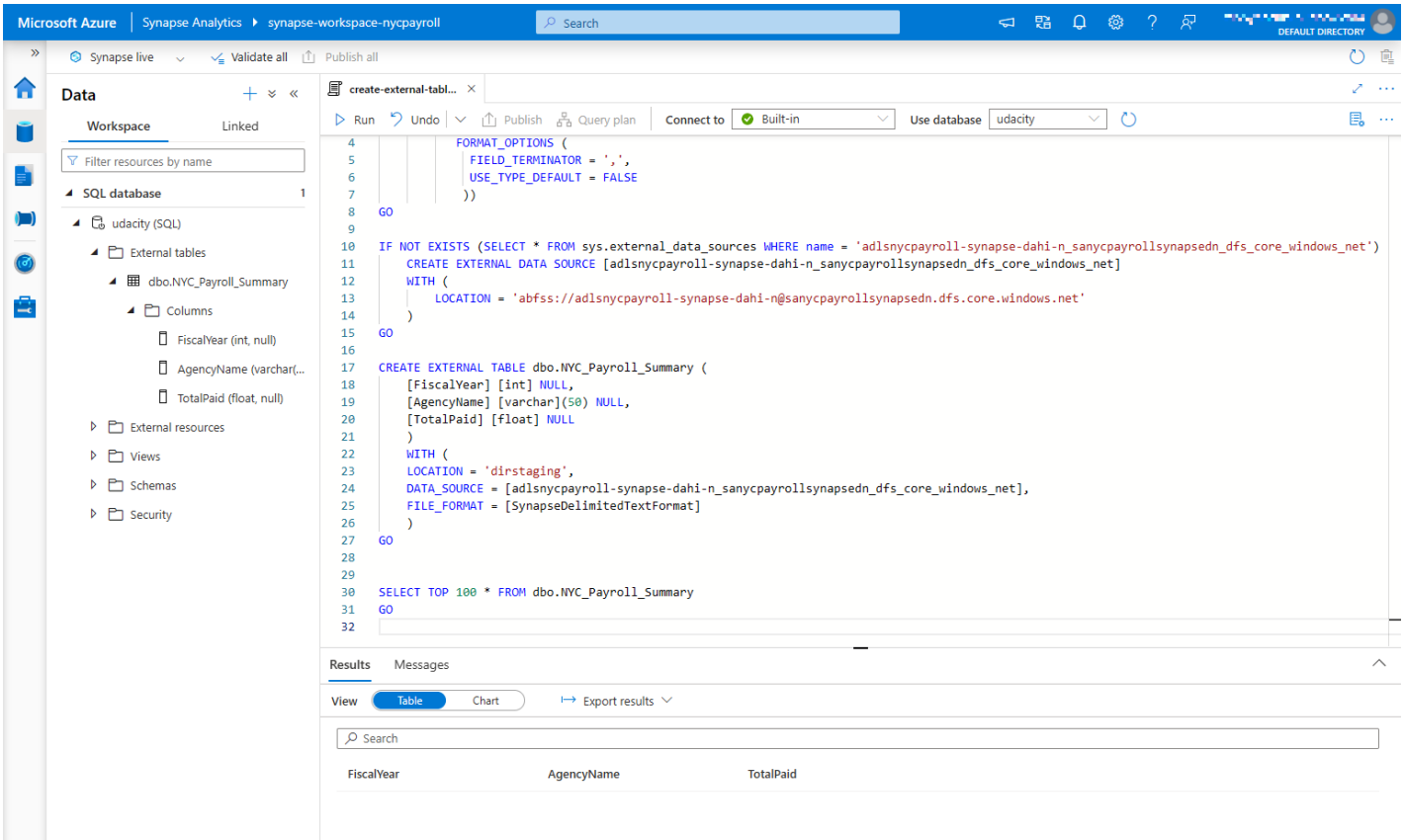
Create a new external data source with the specified name and location. Note that, `adlsnycpayroll-synapse-dahi-n` is the Data Lake Gen 2 storage name, and `sanycpayrollsynapsedn` is the name of the file system (container).

```
IF NOT EXISTS (
    SELECT *
    FROM sys.external_data_sources
    WHERE name = 'adlsnycpayroll-synapse-dahi-n_sanycpayrollsynapsedn_dfs_core_windows_net'
) CREATE EXTERNAL DATA SOURCE [adlsnycpayroll-synapse-dahi-n_sanycpayrollsynapsedn_dfs_core_windows_net]
WITH (
    LOCATION = 'abfss://adlsnycpayroll-synapse-dahi-n@sanycpayrollsynapsedn.dfs.core.windows.net'
```

```
)
GO
```

Create external table that references the `dirstaging` directory of DataLake Gen2 storage for staging payroll summary data. (Pipeline for this will be created in later section)

```
CREATE EXTERNAL TABLE dbo.NYC_Payroll_Summary (
    [FiscalYear] [int] NULL,
    [AgencyName] [varchar](50) NULL,
    [TotalPaid] [float] NULL
) WITH (
    LOCATION = 'dirstaging',
    DATA_SOURCE = [adlsnycpayroll-synapse-dahi-n_sanycpayrollsynapsedn_dfs_core_windows_net],
    FILE_FORMAT = [SynapseDelimitedTextFormat]
)
GO
```



2.6. Create Master and Payroll Transaction Tables in SQL DB

Create Employee Master Data table:

```
CREATE TABLE [dbo].[NYC_Payroll_EMP_MD](
```

```
    [EmployeeID] [varchar](10) NULL,
    [LastName] [varchar](20) NULL,
    [FirstName] [varchar](20) NULL
)
GO
```

Create Job Title Table:

```
CREATE TABLE [dbo].[NYC_Payroll_TITLE_MD](
    [TitleCode] [varchar](10) NULL,
    [TitleDescription] [varchar](100) NULL
)
GO
```

Create Agency Master table:

```
CREATE TABLE [dbo].[NYC_Payroll_AGENCY_MD](
    [AgencyID] [varchar](10) NULL,
    [AgencyName] [varchar](50) NULL
)
GO
```

Create Payroll 2020 transaction data table:

```
CREATE TABLE [dbo].[NYC_Payroll_Data_2020](
    [FiscalYear] [int] NULL,
    [PayrollNumber] [int] NULL,
    [AgencyID] [varchar](10) NULL,
    [AgencyName] [varchar](50) NULL,
    [EmployeeID] [varchar](10) NULL,
    [LastName] [varchar](20) NULL,
    [FirstName] [varchar](20) NULL,
    [AgencyStartDate] [date] NULL,
    [WorkLocationBorough] [varchar](50) NULL,
    [TitleCode] [varchar](10) NULL,
    [TitleDescription] [varchar](100) NULL,
    [LeaveStatusasofJune30] [varchar](50) NULL,
    [BaseSalary] [float] NULL,
    [PayBasis] [varchar](50) NULL,
    [RegularHours] [float] NULL,
    [RegularGrossPaid] [float] NULL,
    [OTHours] [float] NULL,
    [TotalOTPaid] [float] NULL,
    [TotalOtherPay] [float] NULL
)
```

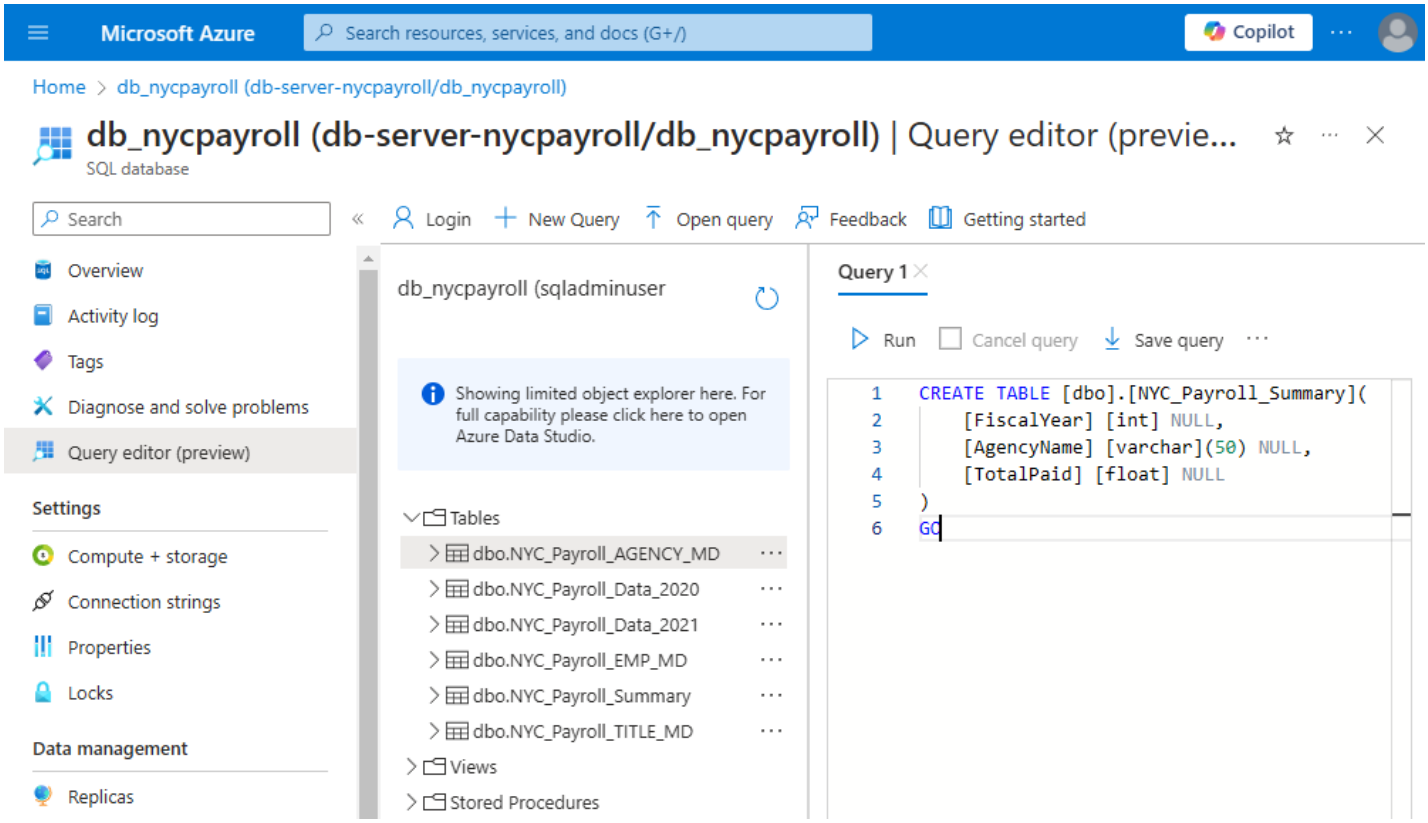
```
GO
```

Create Payroll 2021 transaction data table:

```
CREATE TABLE [dbo].[NYC_Payroll_Data_2021](
    [FiscalYear] [int] NULL,
    [PayrollNumber] [int] NULL,
    [AgencyCode] [varchar](10) NULL,
    [AgencyName] [varchar](50) NULL,
    [EmployeeID] [varchar](10) NULL,
    [LastName] [varchar](20) NULL,
    [FirstName] [varchar](20) NULL,
    [AgencyStartDate] [date] NULL,
    [WorkLocationBorough] [varchar](50) NULL,
    [TitleCode] [varchar](10) NULL,
    [TitleDescription] [varchar](100) NULL,
    [LeaveStatusasofJune30] [varchar](50) NULL,
    [BaseSalary] [float] NULL,
    [PayBasis] [varchar](50) NULL,
    [RegularHours] [float] NULL,
    [RegularGrossPaid] [float] NULL,
    [OTHours] [float] NULL,
    [TotalOTPaid] [float] NULL,
    [TotalOtherPay] [float] NULL
)
GO
```

Create Payroll summary data table:

```
CREATE TABLE [dbo].[NYC_Payroll_Summary](
    [FiscalYear] [int] NULL,
    [AgencyName] [varchar](50) NULL,
    [TotalPaid] [float] NULL
)
GO
```



3. Create Linked Services

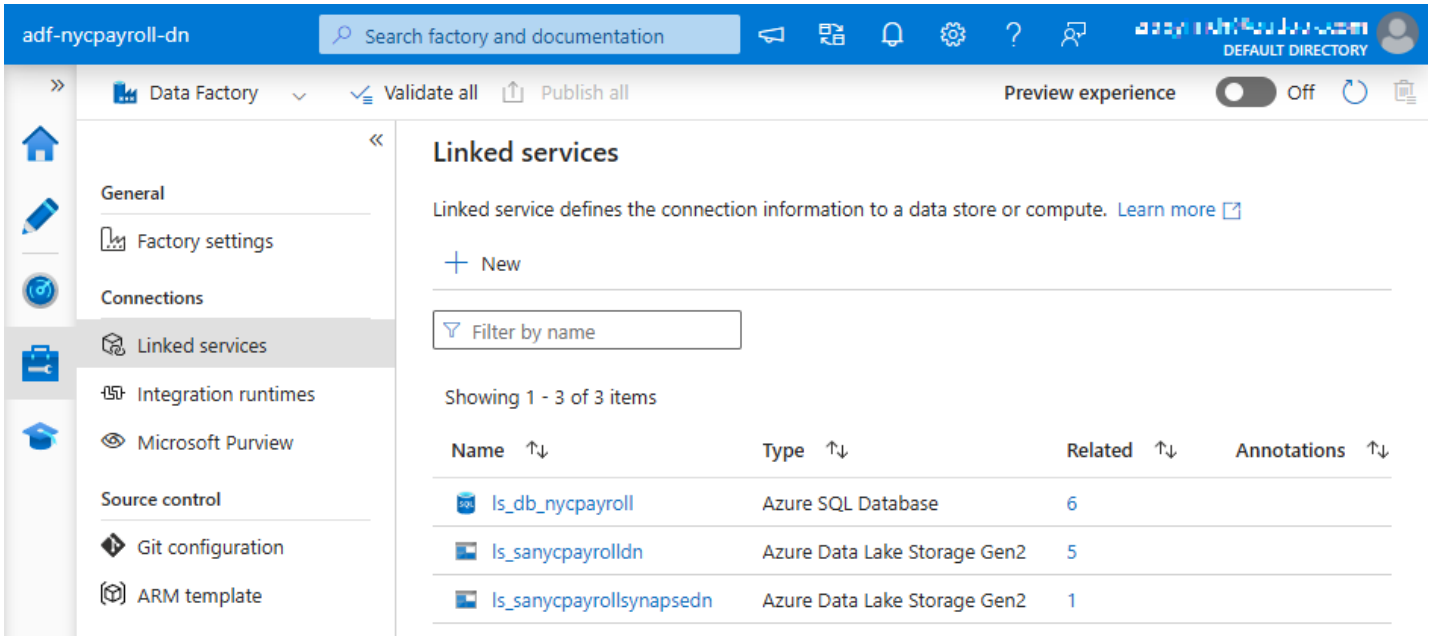
3.1. Create a Linked Service for Azure Data Lake

In Azure Data Factory, create a linked service to the data lake that contains the data files

- From the data stores, select Azure Data Lake Gen 2
- Test the connection

3.2. Create a Linked Service to SQL Database

- If you get a connection error, remember to add the IP address to the firewall settings in SQL DB in the Azure Portal.

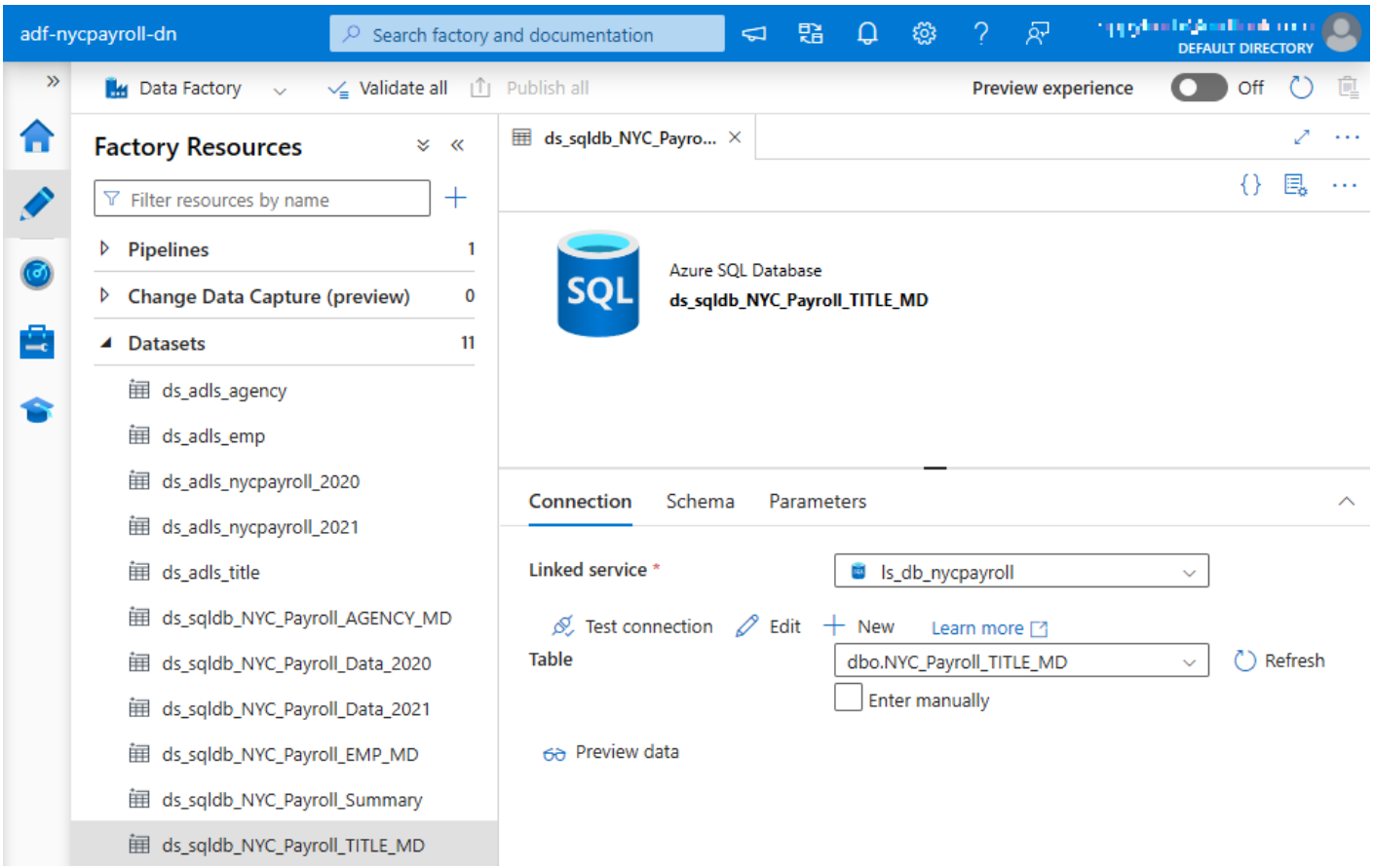


4. Create Datasets

4.1. Create the datasets for the files on Azure Data Lake Gen2

- Select DelimitedText.
- Set the path to nycpayroll_2021.csv in the Data Lake.
- Preview the data to make sure it is correctly parsed.
- Repeat the same process to create datasets for the rest of the data files in the Data Lake:
    - EmpMaster.csv
    - TitleMaster.csv
    - AgencyMaster.csv
- Remember to publish all the datasets

4.2. Create the dataset for all the data tables in SQL DB

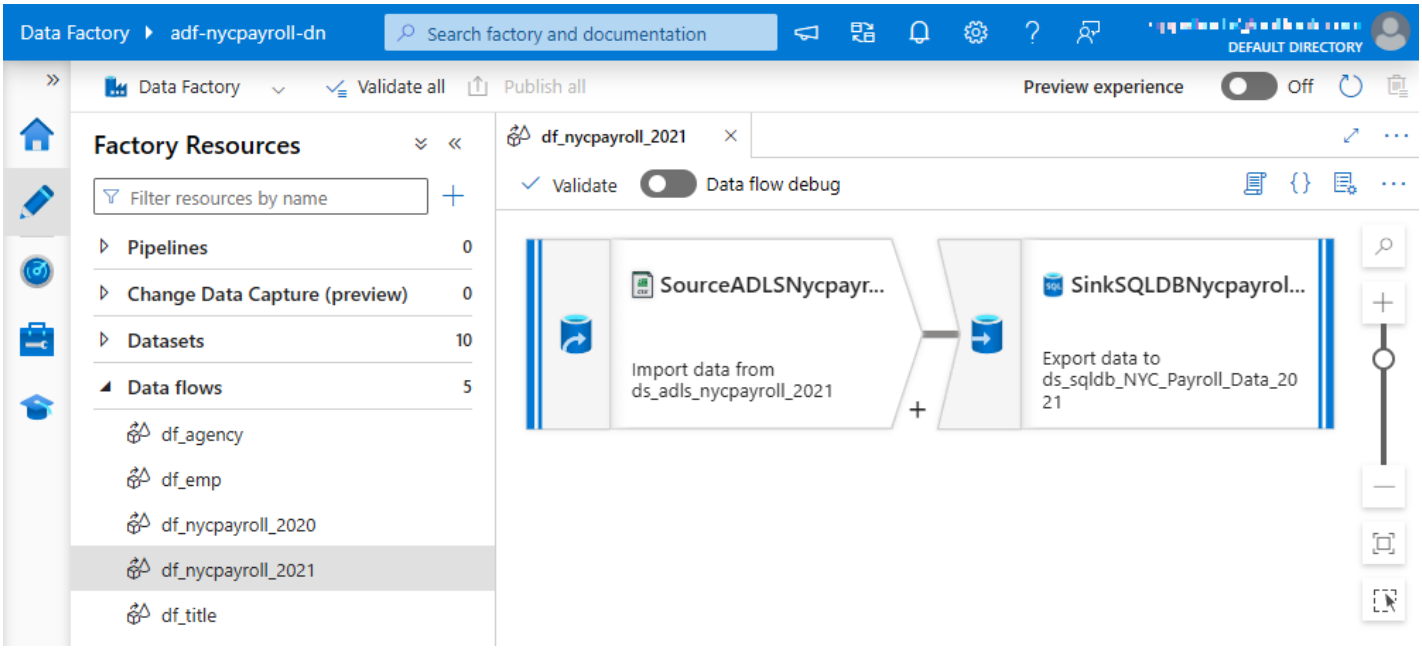


5. Create Data Flows

In Azure Data Factory, create data flow to load 2020 Payroll data from Azure DataLake Gen2 storage to SQL db table created earlier

i. Create a new data flow
ii. Select the dataset for 2020 payroll file as the source
iii. Click on the + icon at the bottom right of the source, from the options choose sink. A sink will get added in the dataflow
iv. Select the sink dataset as 2020 payroll table created in SQL db

Repeat the same process to add data flow to load data for each file in Azure DataLake to the corresponding SQL DB tables.

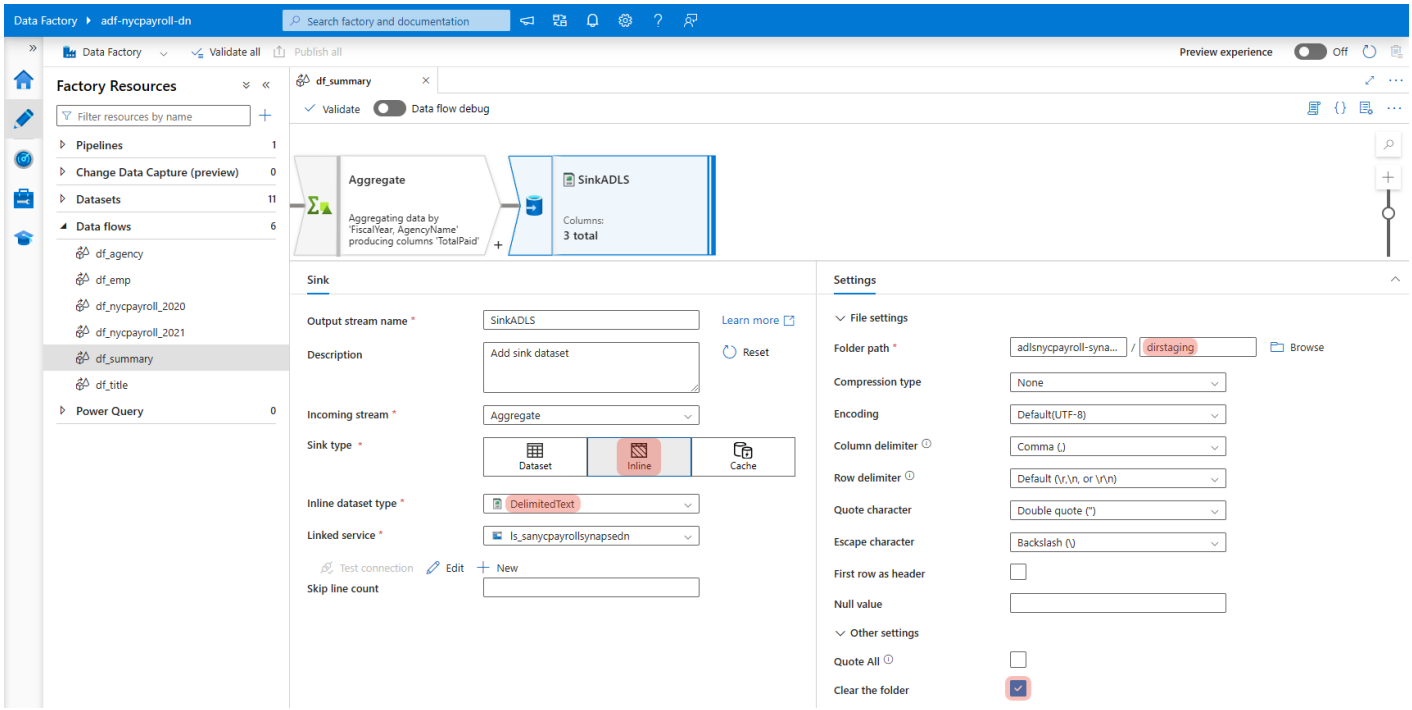


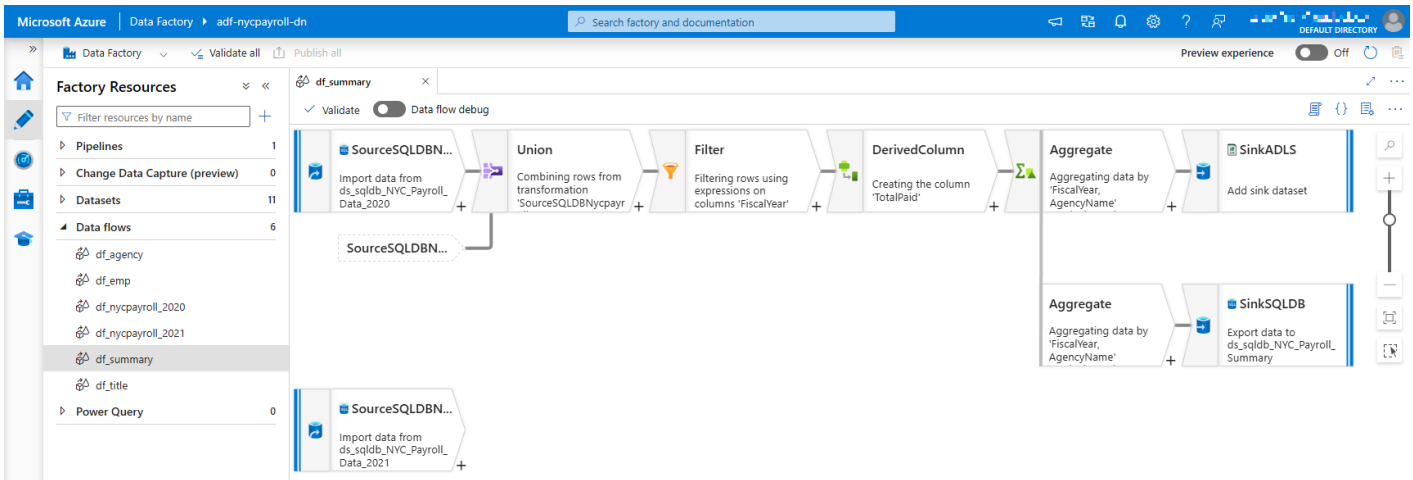
6. Aggregate Data Flow

In this step, we'll extract the 2021 year data and 2020 historical data, merge, aggregate and store it in DataLake staging area which will be used by Synapse Analytics external table. The aggregation will be on `AgencyName`, `FiscalYear` and `TotalPaid`.

i. Create a new data flow and name it df_summary.
ii. Add source as payroll 2020 data from SQL DB.
iii. Add another source as payroll 2021 data from SQL DB.
iv. Create a new Union activity and select both payroll datasets as the source.
v. Make sure to do any source to target mappings if required. This can be done by adding a Select activity before Union.
vi. After Union, add a Filter activity, go to Expression builder.
    - Create a parameter named `dataflow_param_fiscalyear` and give value `2020` or `2021`.
    - Include expression to be used for filtering: `toInteger(FiscalYear) >= $dataflow_param_fiscalyear`.
vii. Now, choose Derived Column after filter.
    - Name the column: `TotalPaid`.
    - Add following expression: `RegularGrossPaid + TotalOTPaid + TotalOtherPay`.
viii. Add an Aggregate activity to the data flow next to the Derived Column activity.
    - Under Group by, select `AgencyName` and `FiscalYear`.
    - Set the expression to `sum(TotalPaid)`.
ix. Add a Sink activity after the Aggregate.
    - Select the sink as summary table created in SQL db.
    - In Settings, tick Truncate table.
x. Add another Sink activity, this will create two sinks after Aggregate.

- Choose Inline as the sink type.
- Choose DelimitedText as inline dataset type.
- In Settings, select the sink as dirstaging in Azure DataLake Gen2 storage.
- Tick Clear the folder.





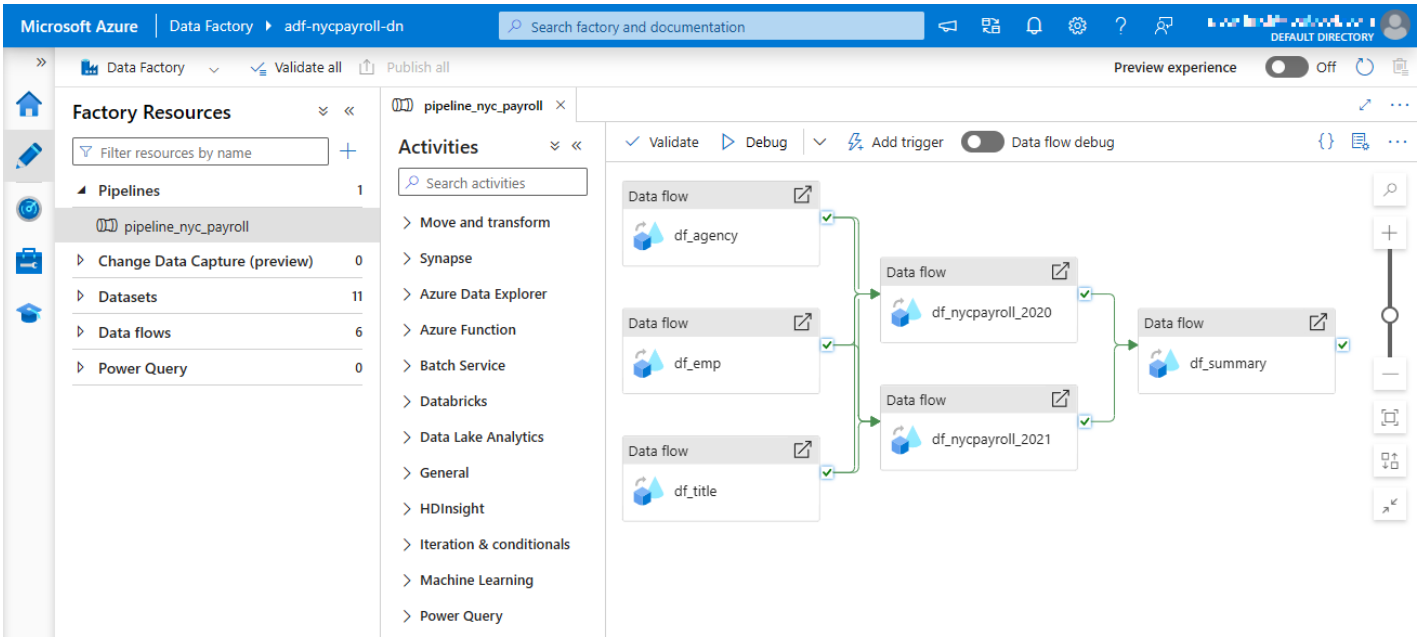
7. Create and Run Pipeline

7.1. Pipeline Creation

Now that we have the data flows created it is time to bring the pieces together and orchestrate the flow.

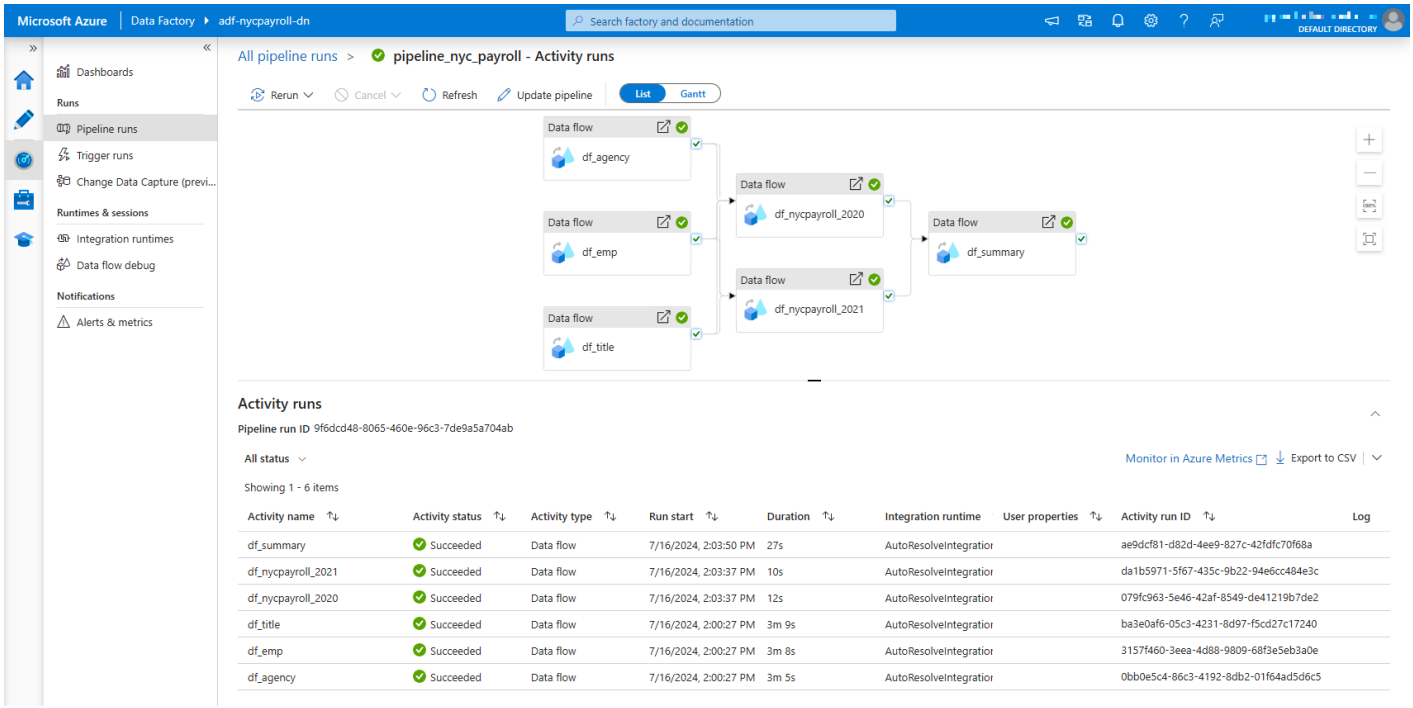
We will create a pipeline to load data from Azure DataLake Gen2 storage in SQL db for individual datasets, perform aggregations and store the summary results back into SQL db destination table and datalake staging storage directory which will be consumed by Synapse Analytics via CETAS.

i. Create a new pipeline.
ii. Include dataflows for Agency, Employee and Title to be parallel.
iii. Add dataflows for payroll 2020 and payroll 2021. These should run only after the initial 3 dataflows have completed.
iv. After payroll 2020 and payroll 2021 dataflows have completed, dataflow for aggregation should be started.
v. Refer to the below screenshot. Your final pipeline should look like this.



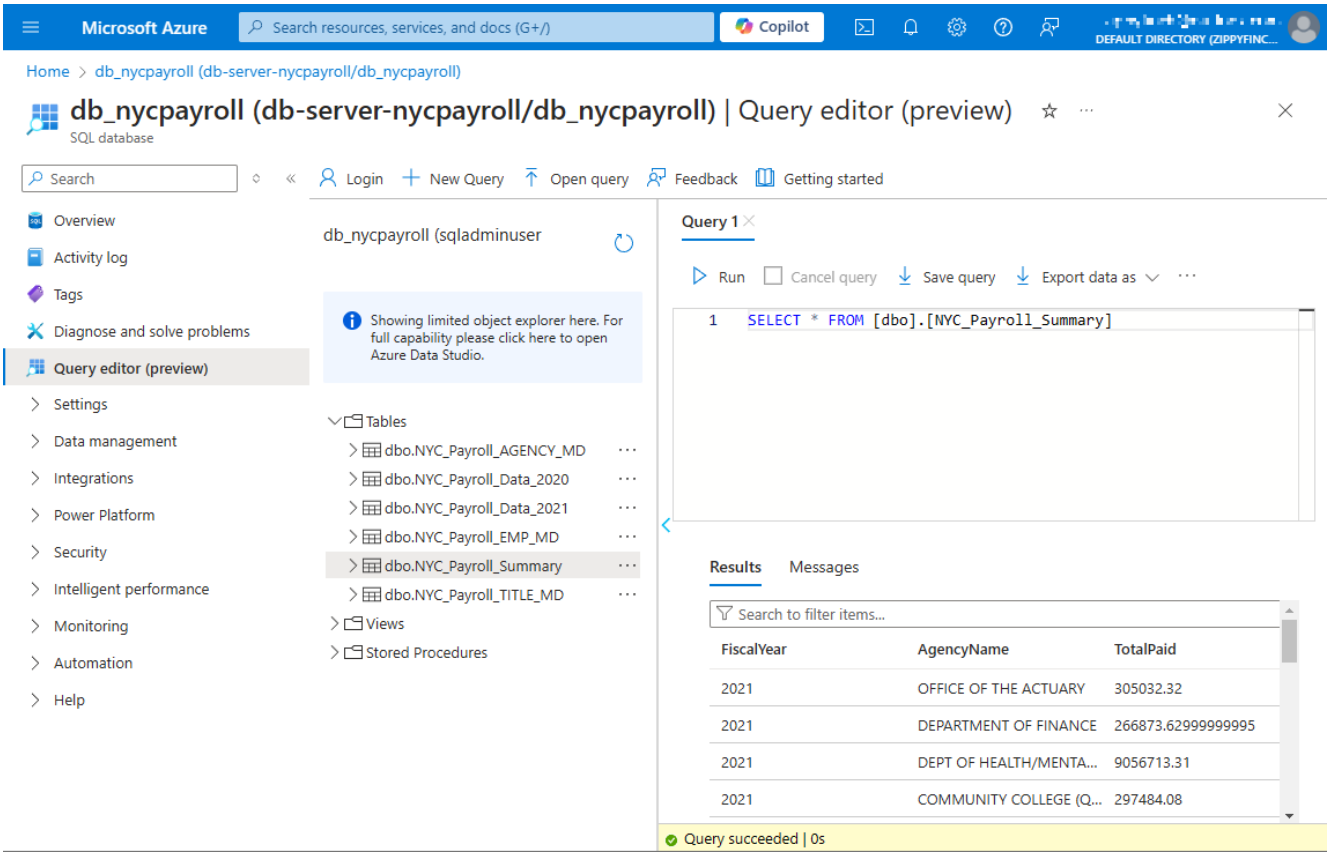
7.2. Trigger and Monitor Pipeline

i. Select Add trigger option from pipeline view in the toolbar
ii. Choose trigger now to initiate pipeline run
iii. You can go to monitor tab and check the Pipeline Runs
iv. Each dataflow will have an entry in Activity runs list

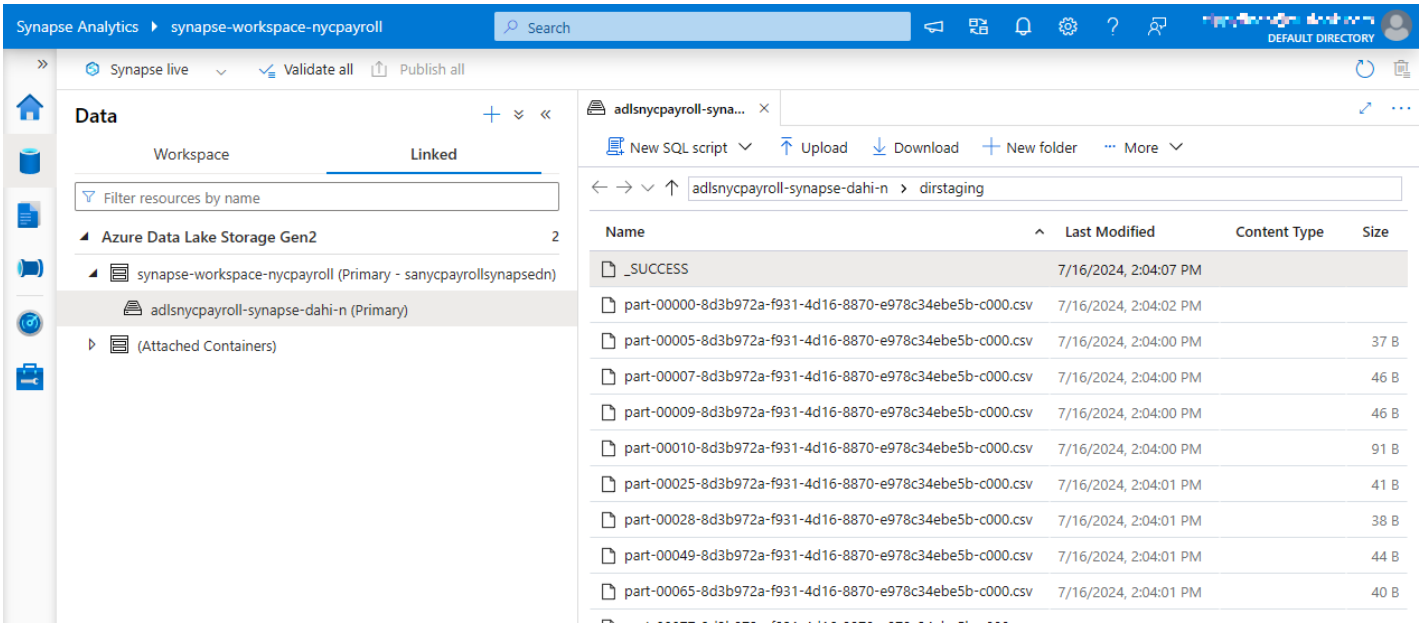


7.3. Verify Pipeline Run Artifacts

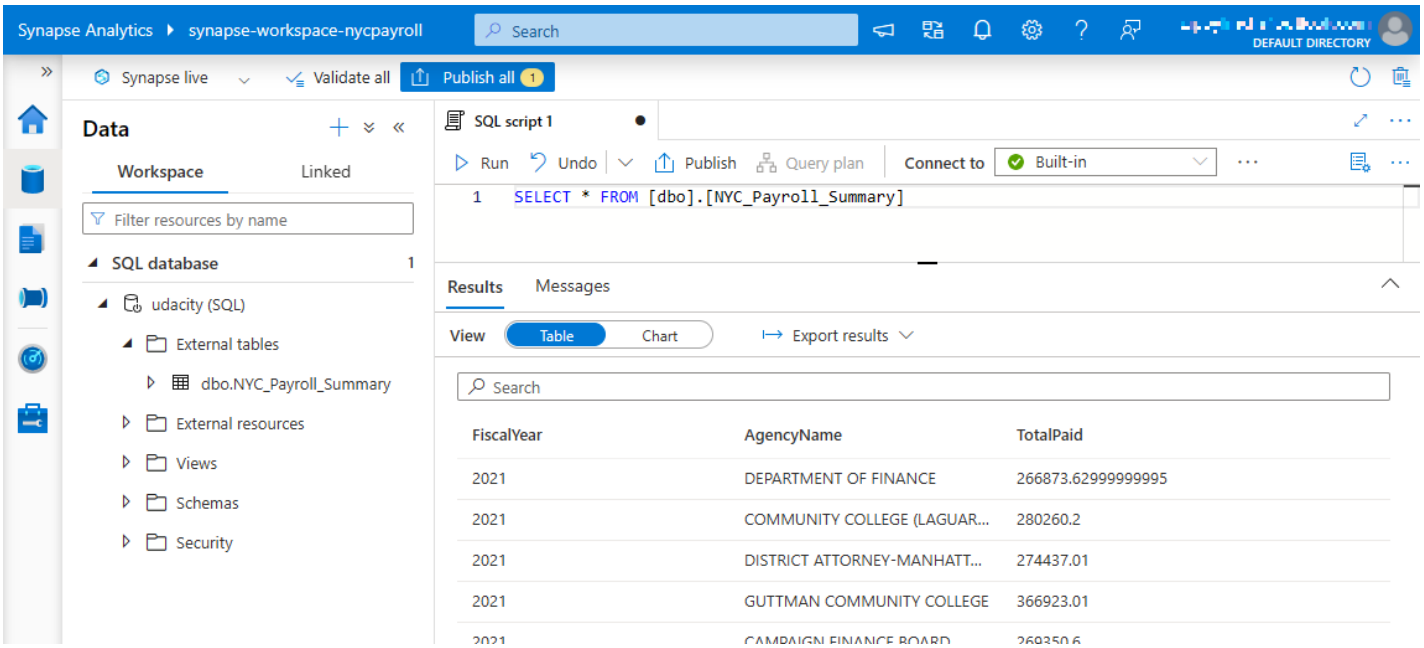
i. Query data in SQL DB summary table (destination table). This is one of the sinks defined in the pipeline.



ii. Check the dirstaging directory in Datalake if files got created. This is one of the sinks defined in the pipeline.



iii. Query data in Synapse external table that points to the dirstaging directory in Datalake.



8. Connect Project to Github and Submit

In this step, we'll connect Azure Data Factory to Github

i. Login to your Github account and create a new Repo in Github
ii. Connect Azure Data Factory to Github
iii. Select your Github repository in Azure Data Factory

iv. Publish all objects to the repository in Azure Data Factory