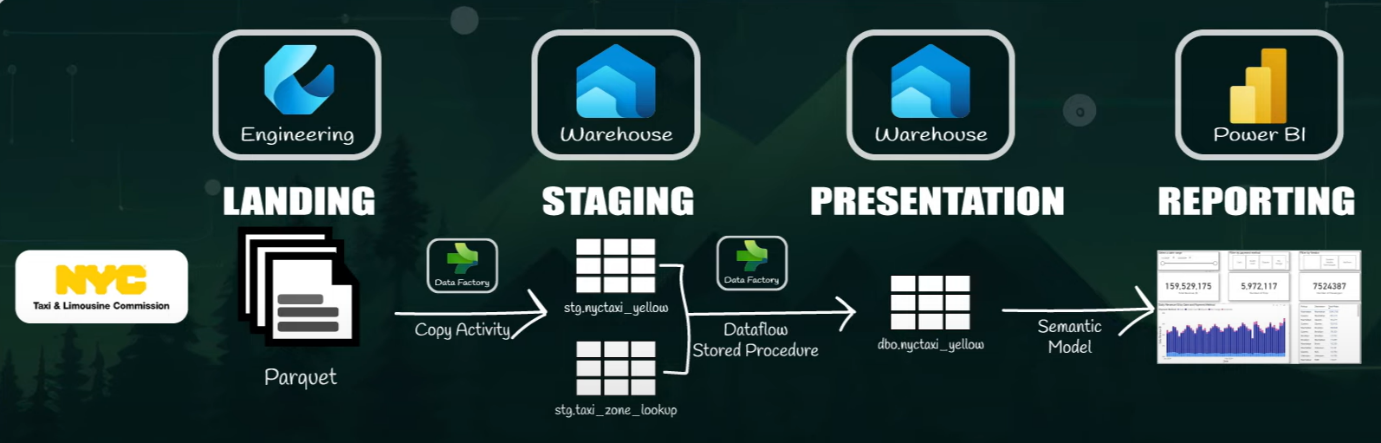
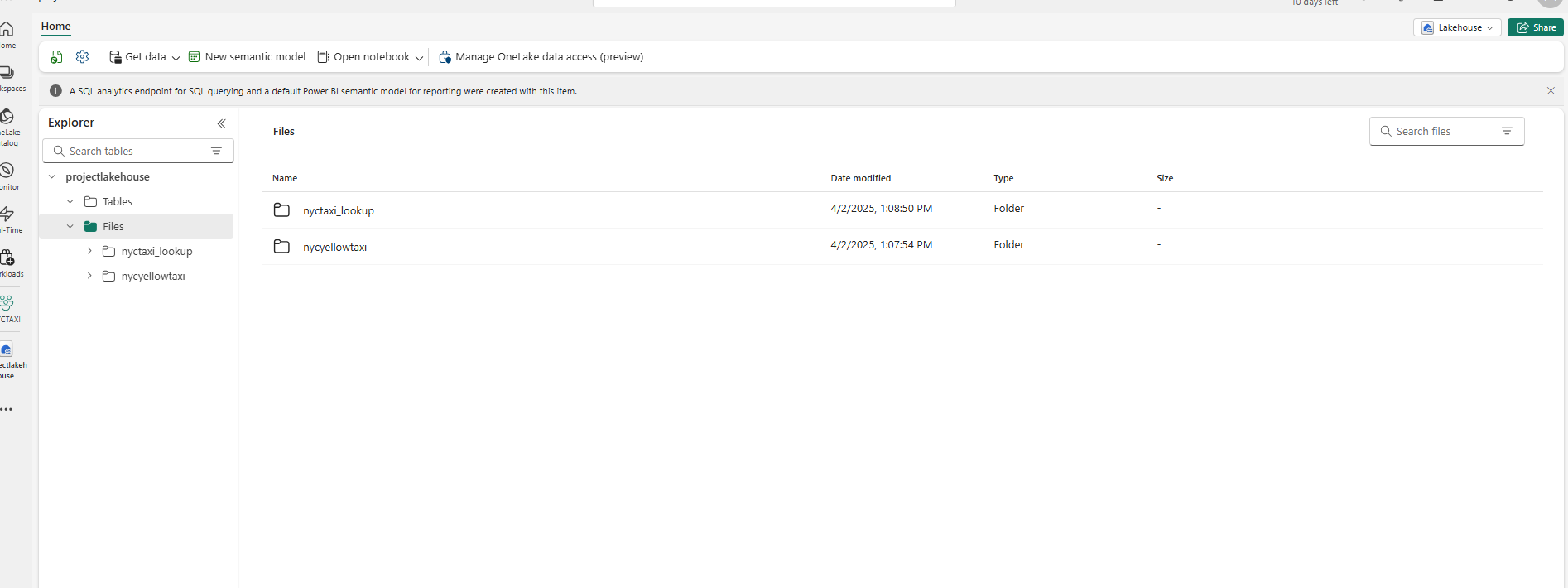
NYC Taxi Trip Data Processing & Reporting

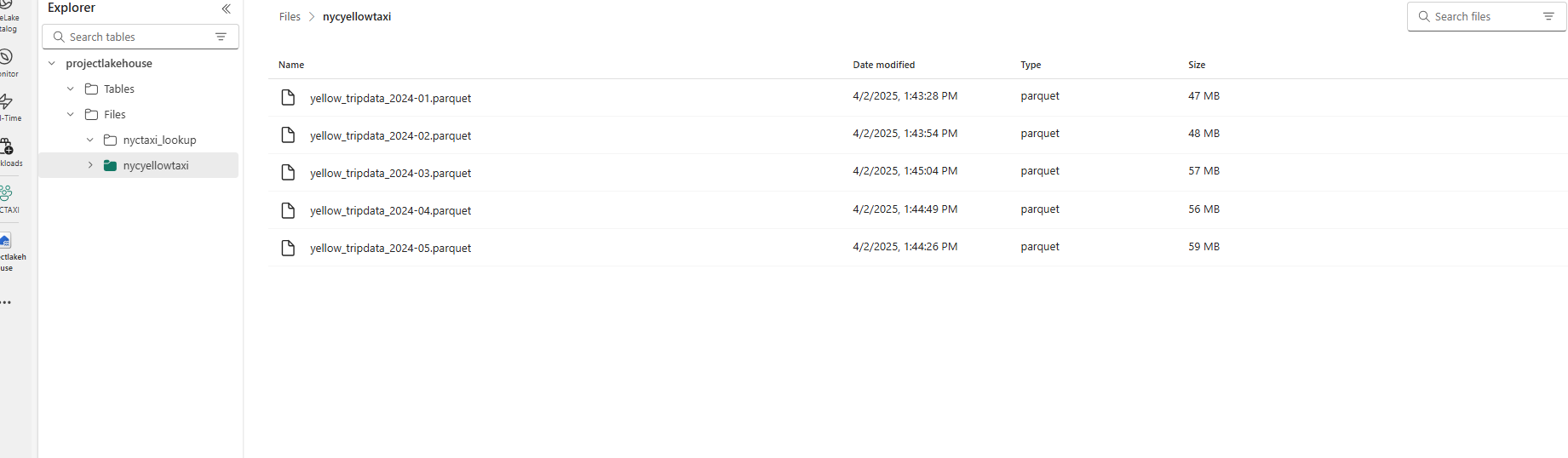
1. **Landing**
   * Source: NYC TLC trip record data in Parquet format.
   * Storage: Data is landed in a Fabric Data Lakehouse.
2. **Staging**
   * Transformation: Azure Data Factory CopyActivity loads data into staging tables (e.g., stg.nyctaxi\_yellow and stg.taxi\_zone\_lookup).
3. **Presentation**
   * Processing: Dataflows and StoredProcedures refine and aggregate the data into the presentation layer (dbo.nyctaxi\_yellow).
4. **Reporting**
   * Visualization: The cleaned and processed data is used to create a SemanticModel, which feeds into Power **BI** Dashboards for analytics and reporting.





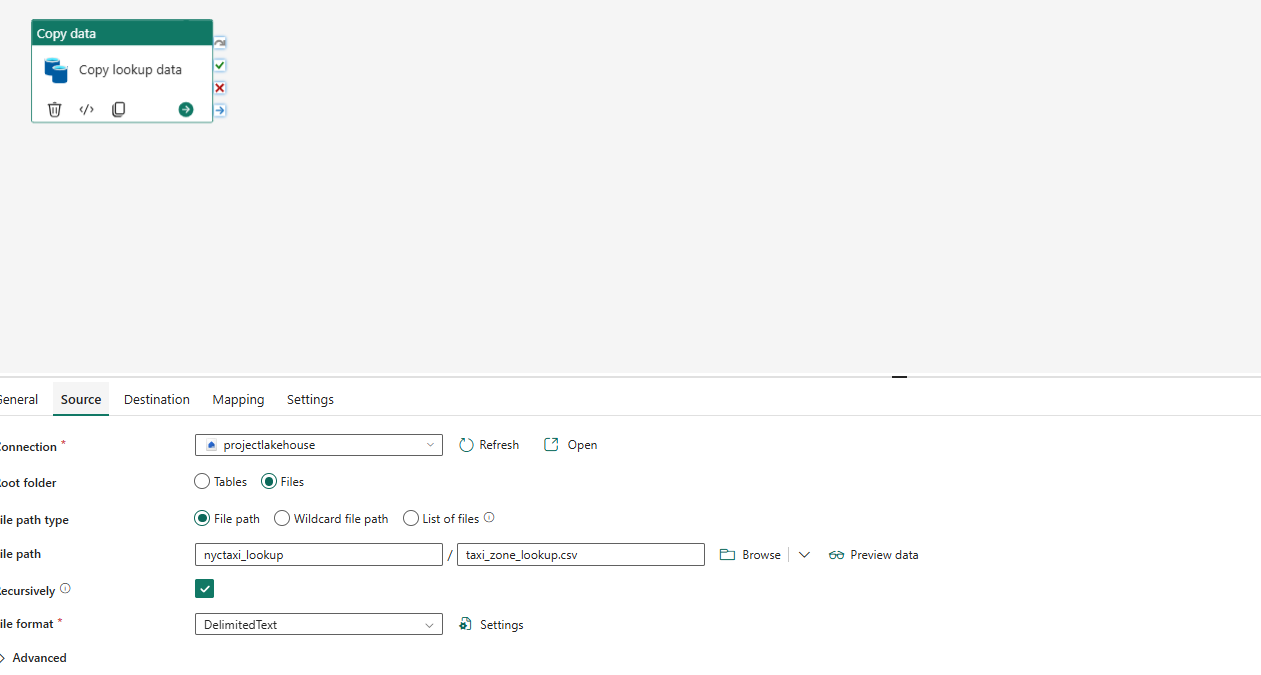
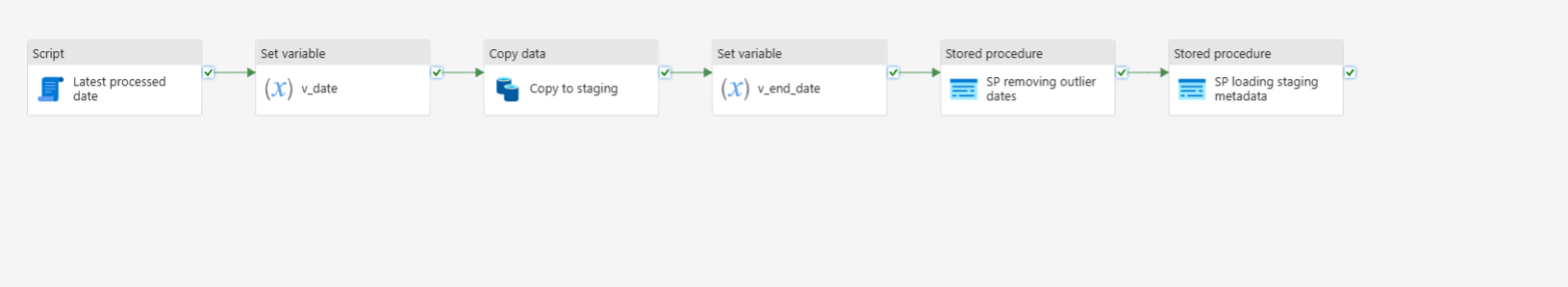
**Landing**

* **Source**: NYC TLC trip record data in **Parquet** format.
* **Storage**: Data is landed in a Fabric Data Lakehouse.



In fabric lakehouse where all month taxi trip data contains trip distance trip mode pickup date

Drop off time payment mode vendor id total passenger

  
in copy data we move data from lakehouse external file to warhouse table   
  
  
  


We set a **dynamic parameter** to store the variable v\_date as @formatDateTime(addToTime(activity('Latest Processed Date').output.resultSets[0].rows[0].latest\_processed\_pickup, -1, 'Month'), 'yyyy-MM') so that v\_date represents the **previous month**, and in the **Copy Activity** for staging nyc\_yellowtaxi, we pass this parameter in the **source file name and path** as @concat('yellow\_tripdata\_', variables('v\_date'), '.parquet') to dynamically select the correct file, while the **destination table** in the warehouse is stg.nyctaxi\_yellow, and we store another variable @addToTime(concat(variables('v\_date'), '-01'), 1, 'Month') to set the **next month as v\_end\_date**, followed by executing a **Stored Procedure (SP) to remove outliers** from the dataset.

create procedure stg.data\_cleaning\_stg

@end\_date datetime2,

@start\_date datetime2

As delete from stg.nyctaxi\_yellow where tpep\_pickup\_datetime < @start\_date or tpep\_pickup\_datetime > @end\_date;  
this procedure will delete pickuptime nextmonth and end date should less than that month

In sp meta data we create sp   
CREATE PROCEDURE metadata.insert\_staging\_metadata

@pipeline\_run\_id VARCHAR(255),

@table\_name VARCHAR(255),

@processed\_date DATETIME2

AS

INSERT INTO metadata.processing\_log (pipeline\_run\_id, table\_processed, rows\_processed, latest\_processed\_pickup, processed\_datetime)

SELECT

@pipeline\_run\_id AS pipeline\_id,

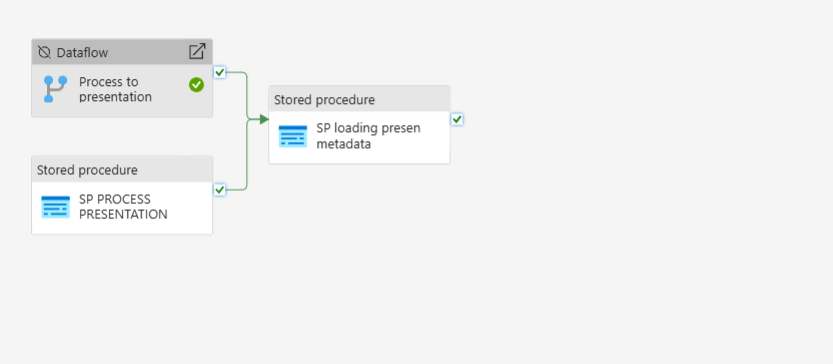
@table\_name AS table\_processed,

COUNT(\*) AS rows\_processed,

MAX(tpep\_pickup\_datetime) AS latest\_processed\_pickup,

@processed\_date AS processed\_datetime

FROM stg.nyctaxi\_yellow;   
we insert column pipeline\_run\_id, table\_processed, rows\_processed, latest\_processed\_pickup, processed\_datetime and store the value



In sp process presentation   
CREATE PROCEDURE dbo.process\_presentation

AS

INSERT INTO dbo.nyctaxi\_yellow

SELECT

CASE

WHEN nty.VendorID = 1 THEN 'Creative Mobile Technologies'

WHEN nty.VendorID = 2 THEN 'VeriFone'

else 'Unknown'

end as vendor,

format(nty.tpep\_pickup\_datetime,'yyyy-MM-dd') as tpep\_pickup\_datetime,

format(nty.tpep\_dropoff\_datetime,'yyyy-MM-dd') as tpep\_dropoff\_datetime,

lu1.Borough as pu\_borough,

lu1.Zone as pu\_zone,

lu2.Borough as pu\_borough,

lu2.Zone as pu\_zone,

CASE

WHEN nty.payment\_type = 1 THEN 'Credit Card'

WHEN nty.payment\_type = 2 THEN 'Cash'

WHEN nty.payment\_type = 3 THEN 'No Charge'

WHEN nty.payment\_type = 4 THEN 'Dispute'

WHEN nty.payment\_type = 5 THEN 'Unknown'

WHEN nty.payment\_type = 6 THEN 'Voided Trip'

else 'Unknown'

end as payment\_method,

nty.passenger\_count as passenger\_count,

nty.trip\_distance as trip\_distance,

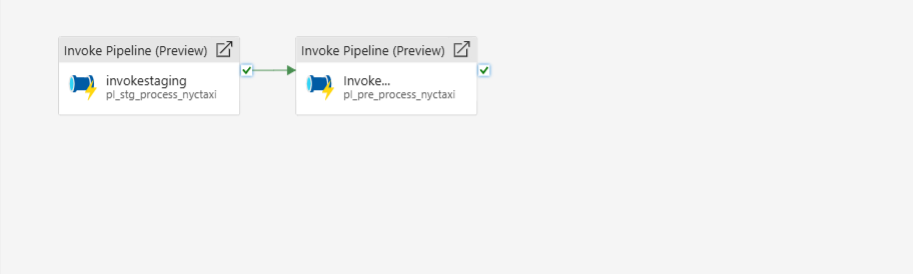
nty.total\_amount as total\_amount

from stg.nyc\_taxi\_yellow nty

left join stg.taxi\_zone\_lookup lu1

on nty.PULocationID = lu1.LocationID

left join stg.taxi\_zone\_lookup lu2

on nty.DOLocationID = lu2.LocationID;  
 and den do some sort of join and change numeric value to assign categorical name and den connected to sp loading metadata   
  


In the invoke staging step, we call the stored procedure pl\_stg\_process\_nyctaxi and pass the data to pl\_pre\_process\_nyctaxi, enabling continuous monthly data fetching from the Lakehouse to the warehouse, which is connected to a semantic model for building Power BI reports.warehouse, which is connected to a semantic model for building Power BI reports.