Copying Data Using Azure Data Factory

Exercise 0 – Setup Azure Data Lake Gen2 Account

1. Complete Lab 1 – Working with Azure Data Lake Gen2 account

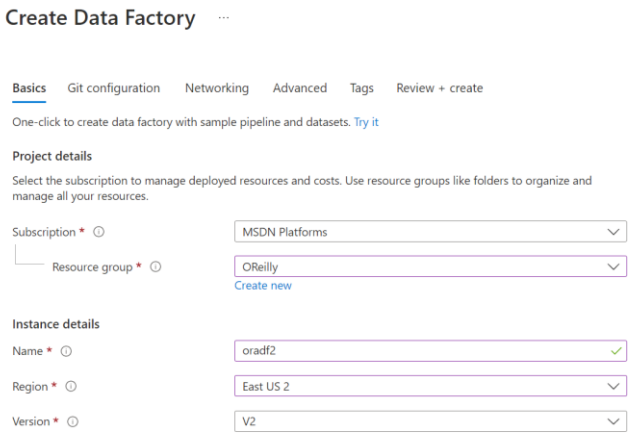
Exercise 1 – Setup Azure Data Factory

1. Go to Azure portal (portal.azure.com)

2. In the search bar, search for Data Factories. And select it

3. Click on Create New

4. Fill up the properties to create account
    a. [Basics Tab]
        i. Select subscription
        ii. Select resource group
        iii. Provide a unique name
        iv. Select region of your choice (example – East US 2)
        v. Select version as V2
        vi. Click Review + Create

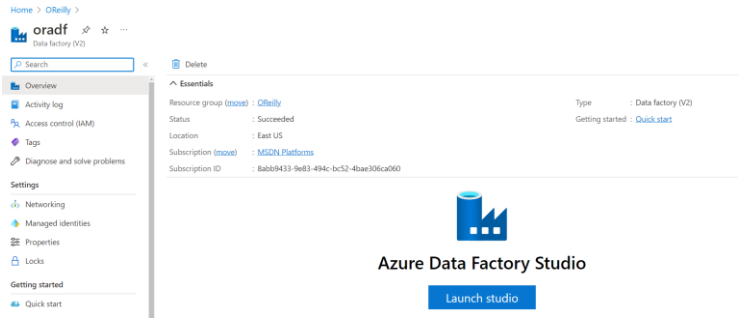


    b. Click Create

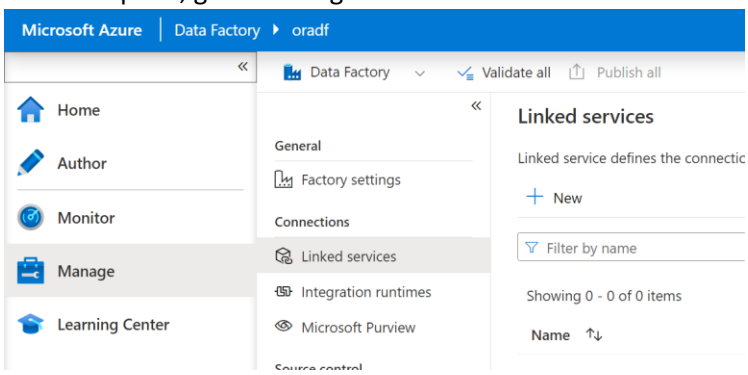
Exercise 2 – Copy File from One Data Lake Folder to Another Using ADF

1. Open Azure Data Factory instance created in the previous step

2. Click on Launch Studio, to open ADF UI

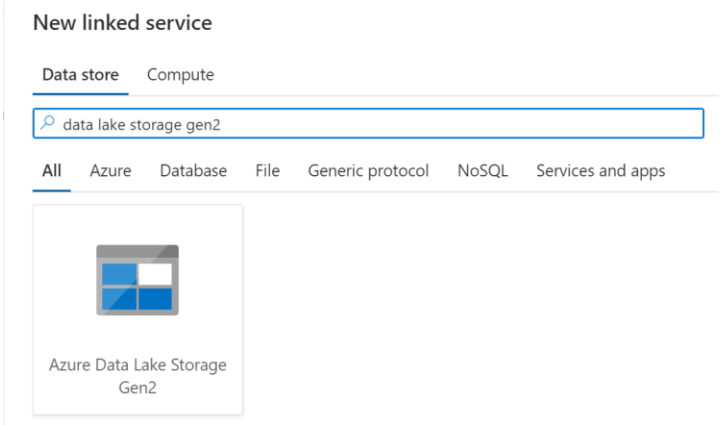


3. From left pane, go to Manage tab. And select Linked Services

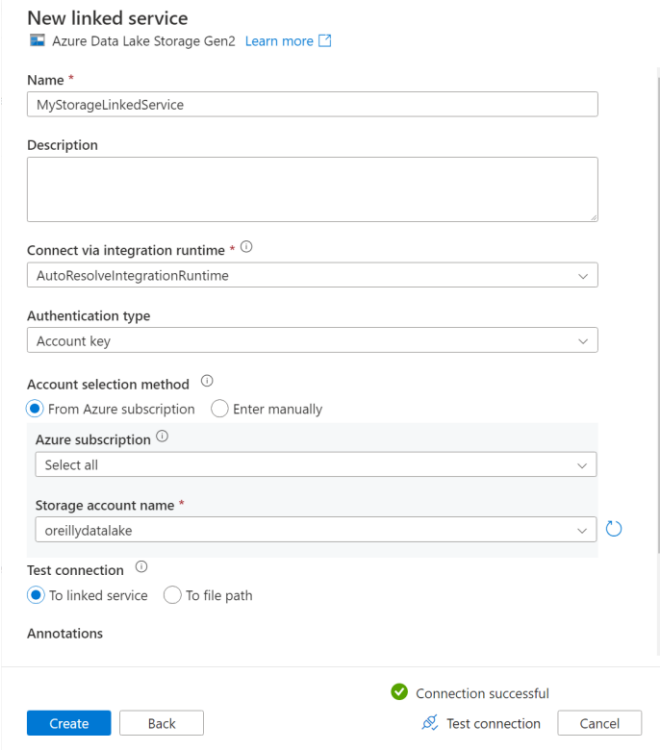


4. Create new Linked Service for Azure Data Lake account
   a. Click New

   b. In New Linked Service window, select Data Store as Azure Data Lake Gen2



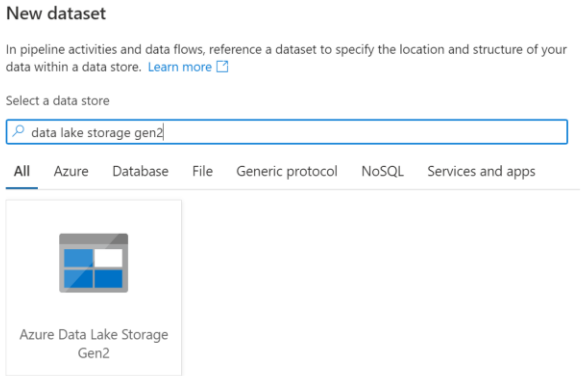
   c. Fill up the properties:
      i. Name: MyStorageLinkedService
      ii. Authentication type: Account key (this is the access key of storage account)
      iii. Storage account name: Select name of Data Lake account
      iv. Click Test connection
      v. If Test connection is successful, click Create



5. Create a dataset for source file – TaxiZones.csv
    a. From left pane, go to Author tab

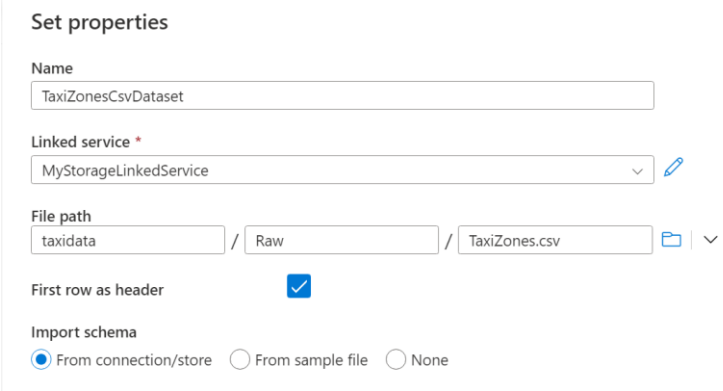
    b. Add a new dataset

    c. In New Dataset window, select Data Store as Azure Data Lake Gen2

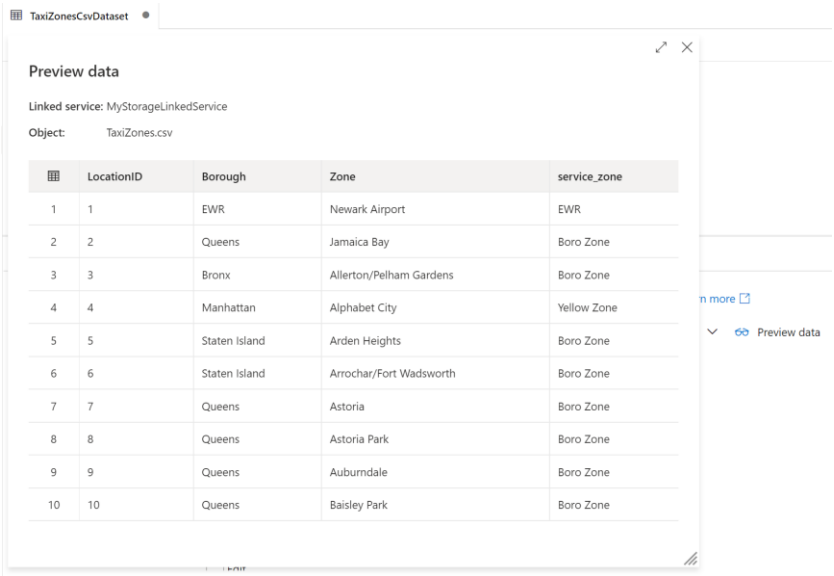


    d. Select format for source file – which is DelimitedText for TaxiZones.csv

    e. Fill up the properties:
        i. Name: TaxiZonesCsvDataset
        ii. Linked service: MyStorageLinkedService
        iii. In file path, select browse. And select TaxiZones.csv file
        iv. Select checkbox: First row as header
        v. Import schema: From connection/store
        vi. Click OK to create



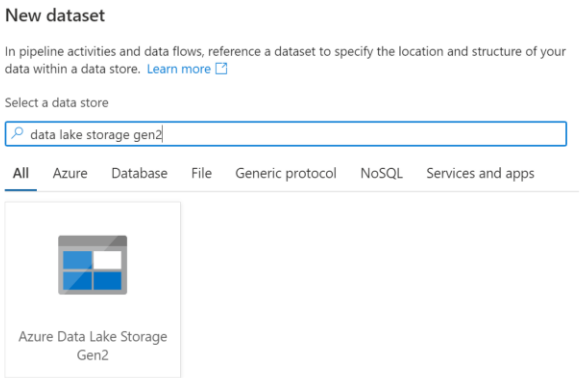
f. Once source dataset is created, click on Preview data to verify there are 4 columns



	LocationID	Borough	Zone	service_zone
1	1	EWR	Newark Airport	EWR
2	2	Queens	Jamaica Bay	Boro Zone
3	3	Bronx	Allerton/Pelham Gardens	Boro Zone
4	4	Manhattan	Alphabet City	Yellow Zone
5	5	Staten Island	Arden Heights	Boro Zone
6	6	Staten Island	Arrochar/Fort Wadsworth	Boro Zone
7	7	Queens	Astoria	Boro Zone
8	8	Queens	Astoria Park	Boro Zone
9	9	Queens	Auburndale	Boro Zone
10	10	Queens	Baisley Park	Boro Zone

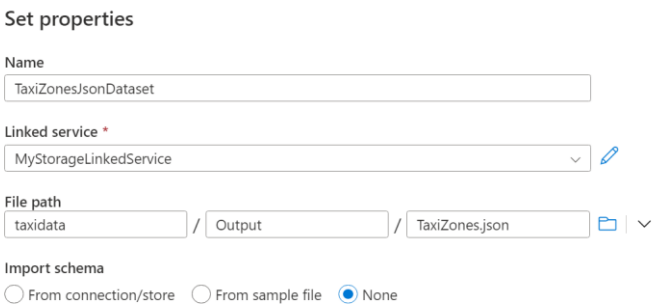
6. Create a dataset for sink file – TaxiZones.json (this file doesn't exist)
   a. Add a new dataset

   b. In New Dataset window, select Data Store as Azure Data Lake Gen2



   c. Select format for sink file – select JSON to store output data in JSON format

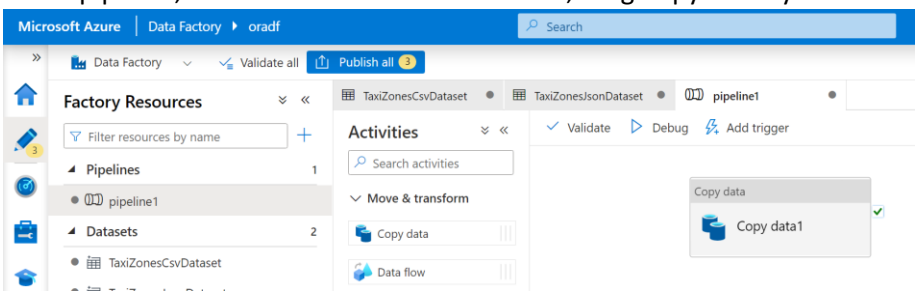
   d. Fill up the properties:
      i. Name: TaxiZonesJsonDataset
      ii. Linked service: MyStorageLinkedService
      iii. In file path, select browse. And select folder location to store output file.
      iv. Manually make changes to path to add folder & file names (see image below)
      v. Select checkbox: First row as header
      vi. Import schema: None (since file doesn't exist, its schema can't be imported)
      vii. Click OK to create



e. After creation, you cannot preview the data since the file doesn't exist

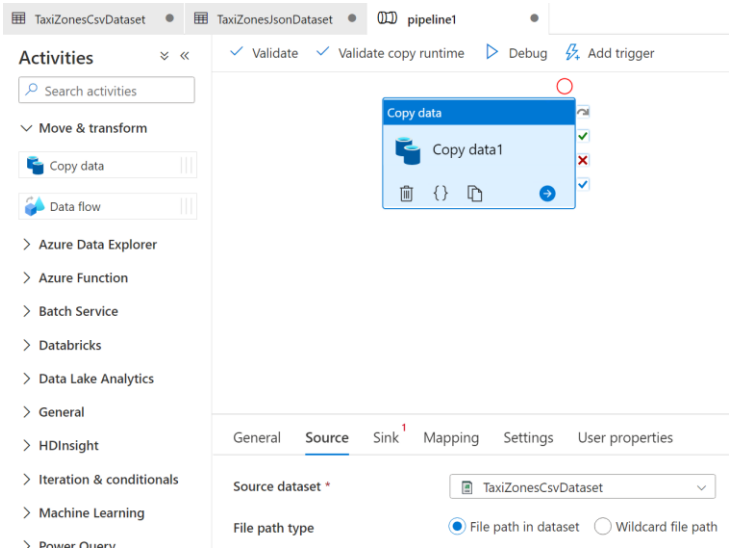
7. Create a pipeline to copy data
   a. In Author tab, create new Pipeline

   b. In the pipeline, from Move & transform section, drag Copy activity on canvas

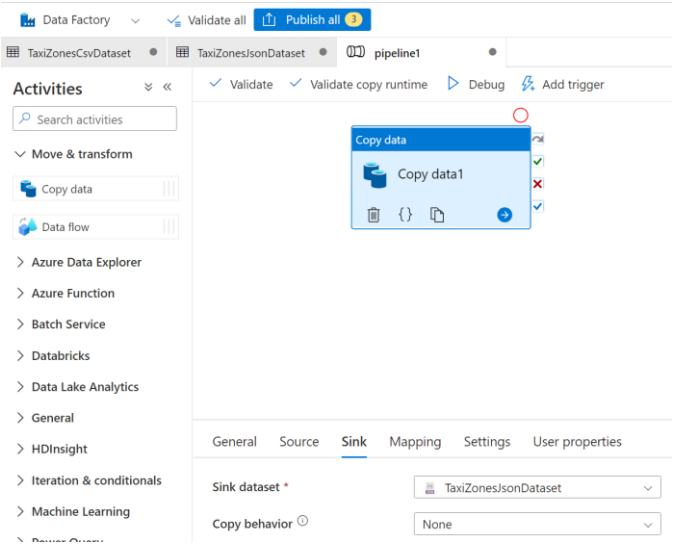


   c. Click on Copy activity

   d. In source tab, select source dataset →TaxiZonesCsvDataset

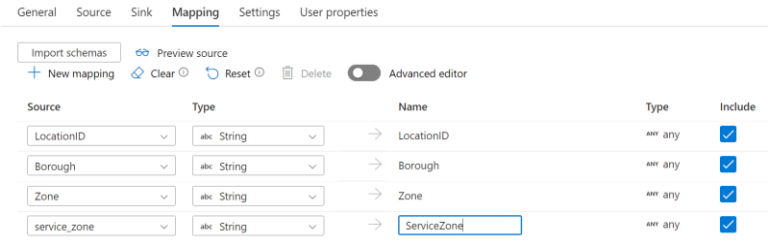


   e. In sink tab, select sink dataset →TaxiZonesJsonDataset



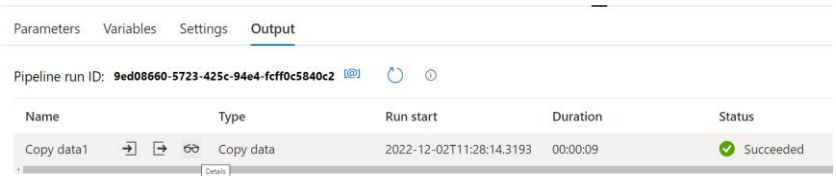
f. In Mapping tab, click Import schemas

g. You can remove or rename sink columns (as shown in image below)



8. Once configured, click on Debug to execute the Pipeline

9. Click anywhere in canvas (but not on CopyData activity). Notice the glasses icon in the pipeline run details. Click on it to see the statistics



10. Go back to Data Lake account to check the creation of new file, TaxiZones.json

11. Click on Publish All to save the changes

Exercise 3 – Setup Azure SQL Database

1. Go to Azure portal (portal.azure.com)

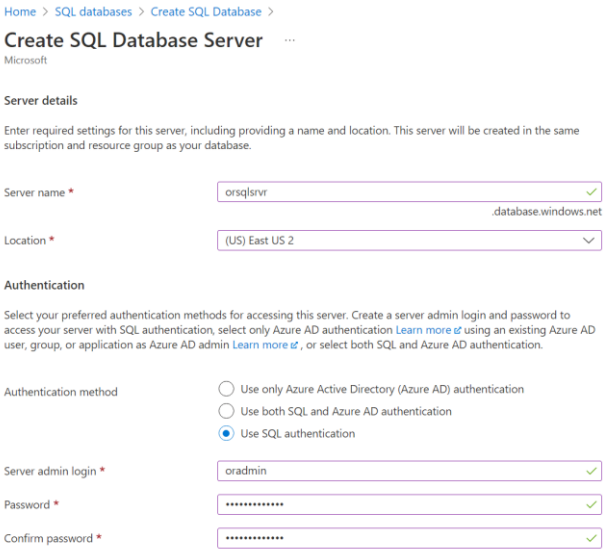
2. In the search bar, search for SQL Databases. And select it

3. Click on Create New

4. Fill up the properties to create Azure SQL database
   a. [Basics Tab]
      i. Select subscription

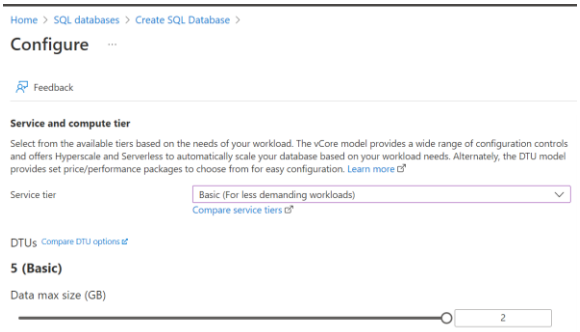
ii. Select resource group
iii. Database name: DB1

iv. Server: Select – Create new
    1. Server name: Provide a unique name
    2. Region: Select a region of your choice
    3. Authentication method: Use SQL authentication
    4. Provide admin username and password
    5. Click create

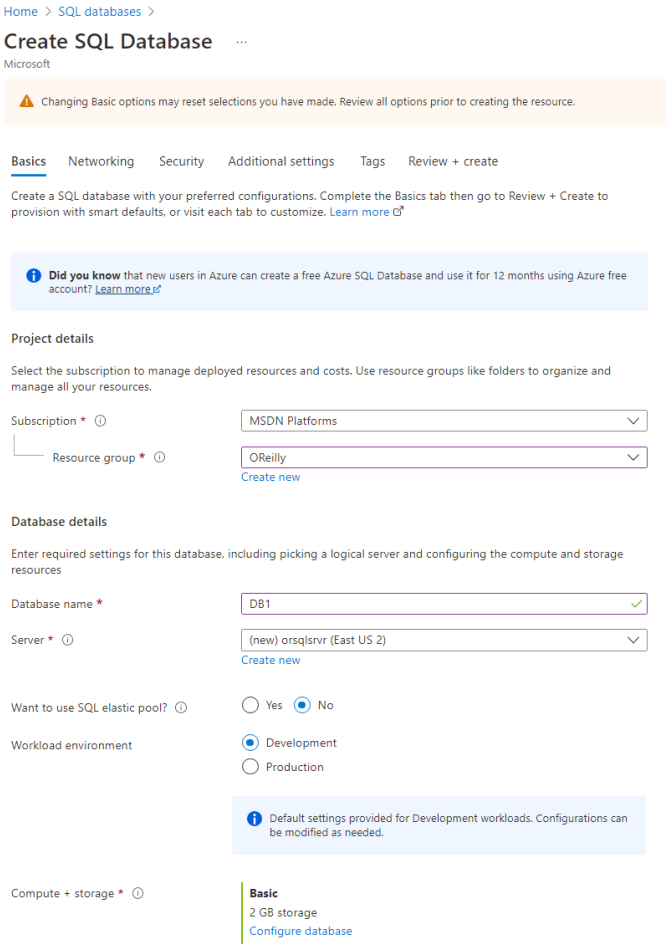


v. Workload environment: Development

vi. Compute + Storage: Click configure database
    1. Service tier: Basic
    2. Apply



vii. See final configuration in image below:

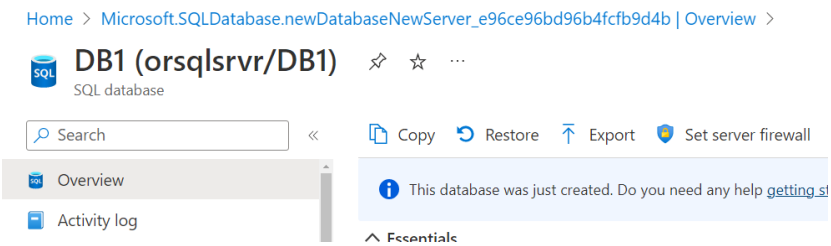


b. [Additional settings Tab]
   i. Use existing data: Sample
   ii. Click Review + Create

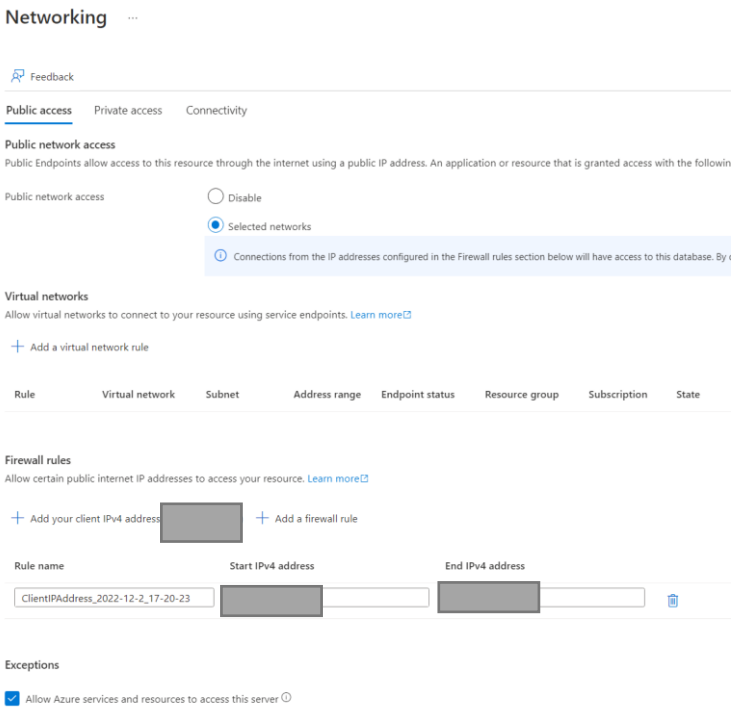
c. Click Create

5. Once created, open Azure SQL database

6. From top menu, select – Set server Firewall



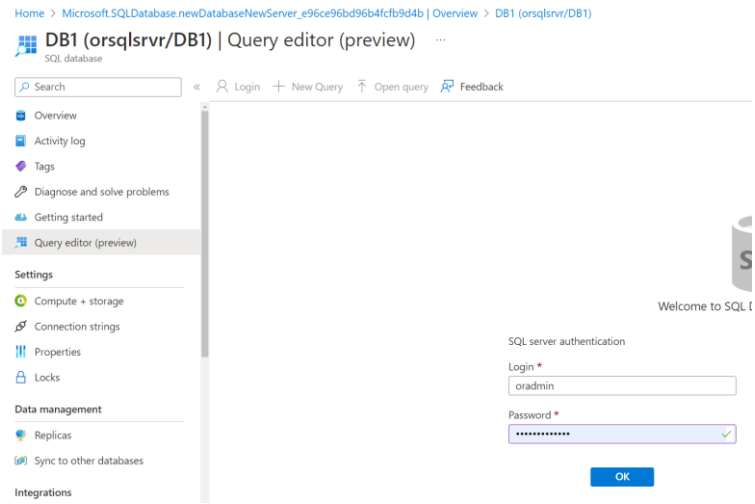
7. In Networking window:
   a. Select Public access
   b. Click on selected networks
   c. In Firewall rules, click on Add your client IPv4 address. This will create a rule as shown below:
   d. In Exceptions, select “Allow Azure services and resources to access this server”



e. Save
f. Close Networking window

8. From left pane of SQL database, go to Query Editor

9. Add your Azure SQL admin username and password that you defined in previous step, and login



10. Once logged in, run the following script to create a watermark table:

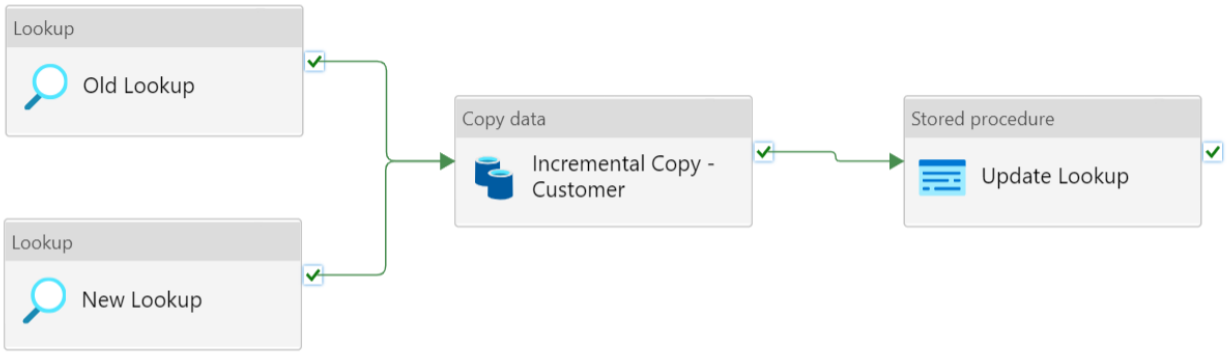
```
CREATE Table Watermark
(
    TableName VARCHAR(255),
    WatermarkValue DATETIME,
);

INSERT INTO Watermark
VALUES ('SalesLT.Customer','2000-01-01');

SELECT * FROM Watermark;
```

11. Remove previous script and run following script to create a stored procedure to update watermark value:

```
CREATE PROCEDURE UpdateWatermark @LastModifiedtime DATETIME, @TableName varchar(50)
```

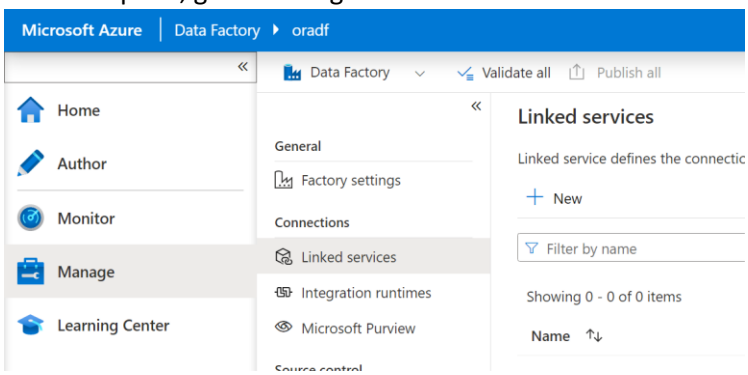
```
AS
BEGIN
    UPDATE Watermark
    SET WatermarkValue = @LastModifiedtime
    WHERE TableName = @TableName
END;
```

Exercise 4 – Copy Data Incrementally from Azure SQL Database to Data Lake



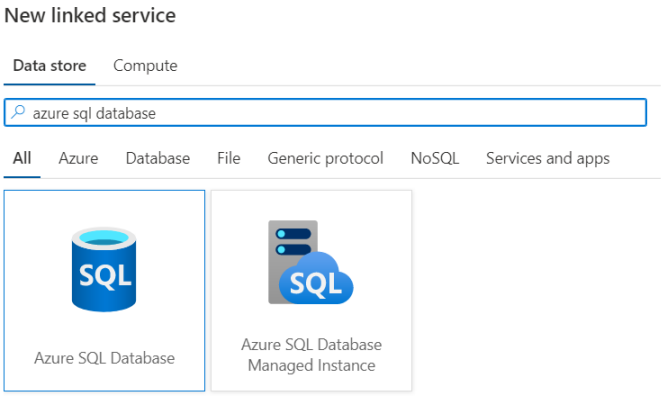
1. Go to Data Factory

2. From left pane, go to Manage tab. And select Linked Services



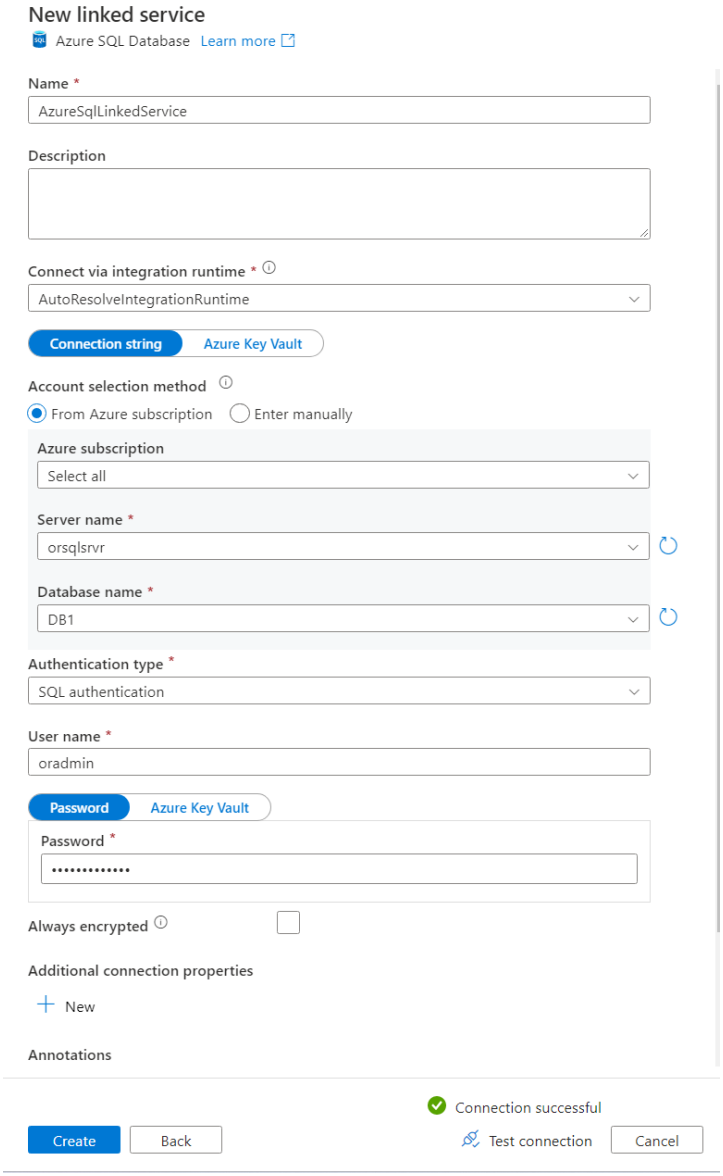
3. Create new Linked Service for Azure SQL database
    a. Click New

    b. In New Linked Service window, select Azure SQL Database



    d. Fill up the properties:
        i. Name: AzureSqlLinkedService

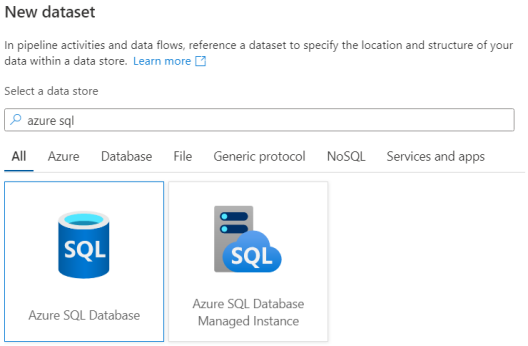
ii. Select your server name
iii. Select your database name
iv. Authentication type: SQL authentication
v. Provide user name & password
vi. Click Test connection
vii. If Test connection is successful, click Create



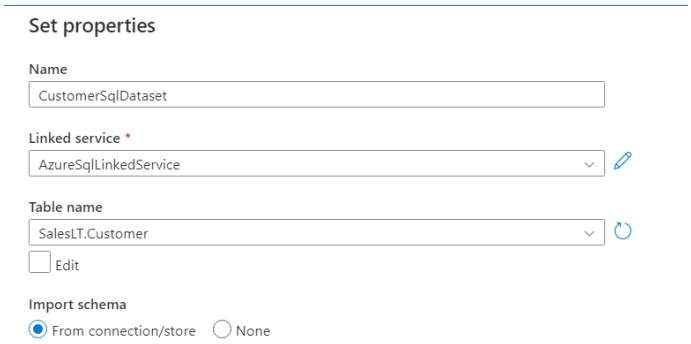
12. Create a dataset for source table – SalesLT.Customer
    a. From left pane, go to Author tab

    b. Add a new dataset

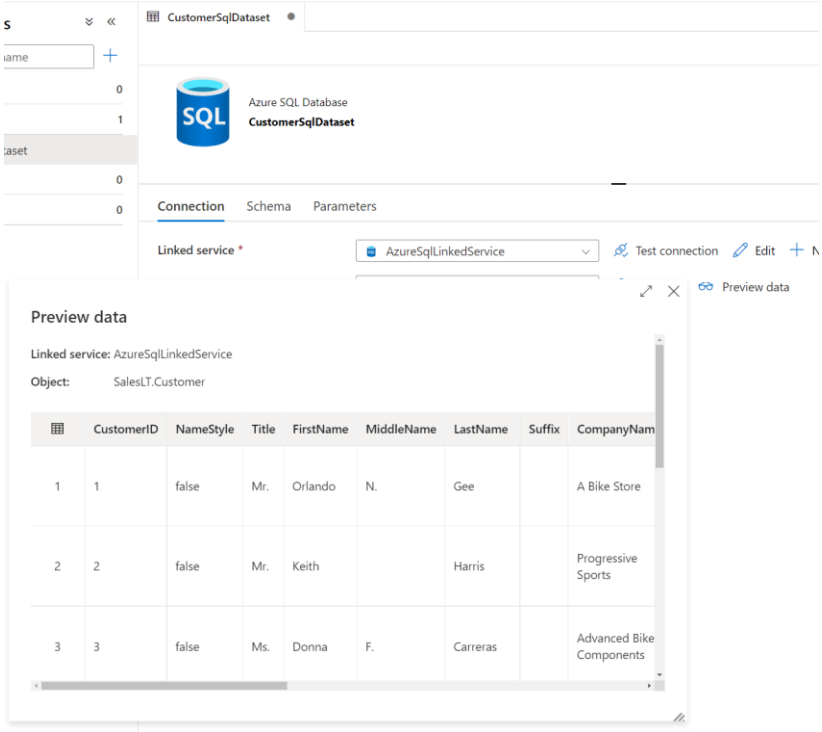
    c. In New Dataset window, select Data Store as Azure SQL Database



d. Fill up the properties:
   i. Name: CustomerSqlDataset
   ii. Linked service: AzureSqlLinkedService
   iii. Select table name: SalesLT.Customer
   iv. Import schema: From connection/store
   v. Click OK to create

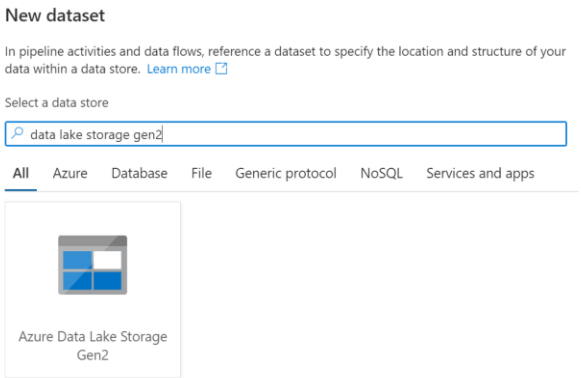


e. Once source dataset is created, click on Preview data to verify table data



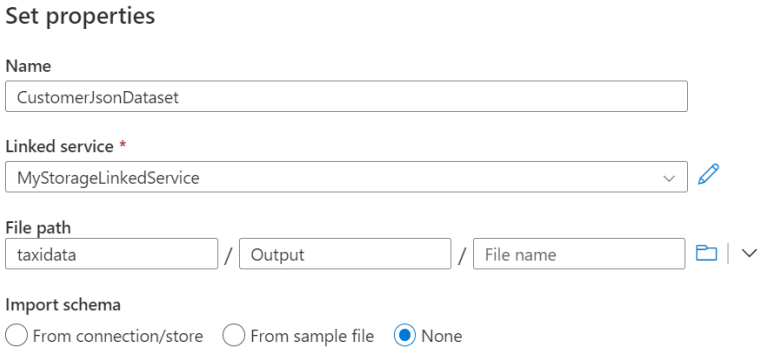
13. Create a dataset for sink file – Customer.json (this file doesn't exist)
   a. Add a new dataset

   b. In New Dataset window, select Data Store as Azure Data Lake Gen2



c. Select format for sink file – select JSON to store output data in JSON format

d. Fill up the properties:
   i. Name: CustomerJsonDataset
   ii. Linked service: MyStorageLinkedService
   iii. In file path, select browse. And select folder location to store output file.
   iv. Manually make changes to path to add folder. Don't add any file name (see image below)
   v. Import schema: None (since file doesn't exist, its schema can't be imported)
   vi. Click OK to create



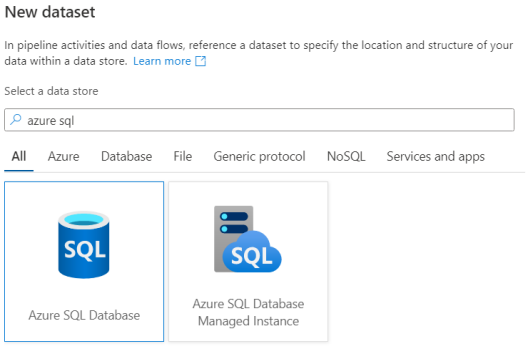
e. After creation, click in File name textbox

f. Click on Add dynamic content, and add the expression to dynamically generate file name:

```
@CONCAT('Customers-', pipeline().RunId, '.json')
```

14. Create a dataset for watermark
   a. Add a new dataset

   b. In New Dataset window, select Data Store as Azure SQL Database

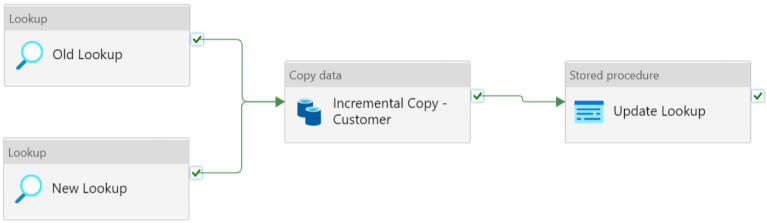


   c. Fill up the properties:
      i. Name: WatermarkSqlDataset
      ii. Linked service: AzureSqlLinkedService
      iii. Select table name: Watermark
      iv. Import schema: From connection/store
      v. Click OK to create

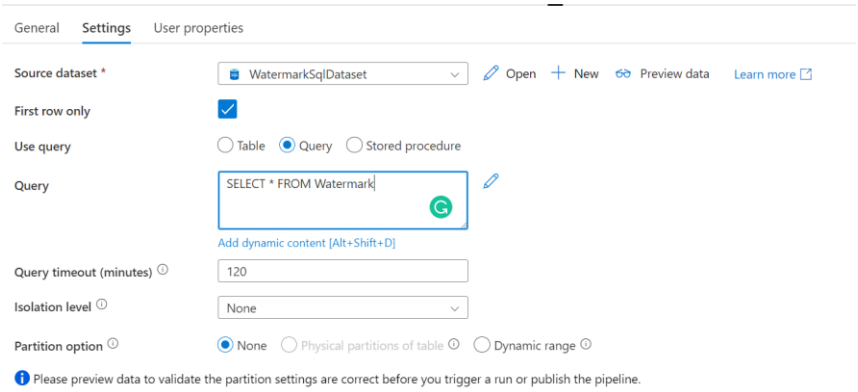
   d. Once watermark dataset is created, click on Preview data to verify table data

15. Create a pipeline to copy data
   a. In Author tab, create new Pipeline

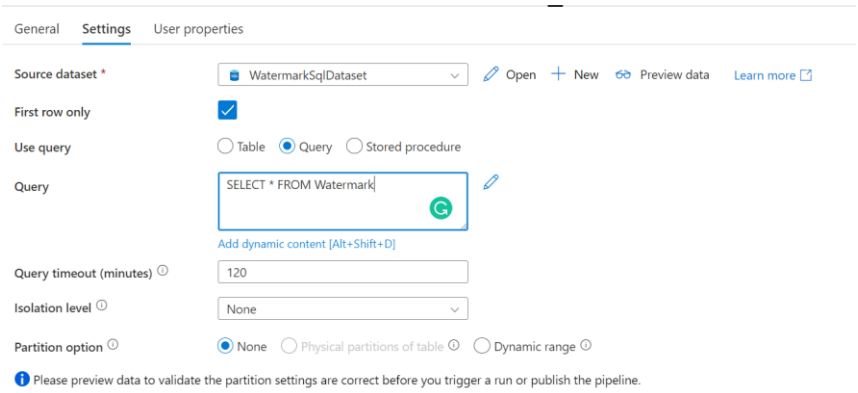
16. In the pipeline, drag two Lookup, one Copy Data and one Stored procedure activities on canvas (see image below). Name them as shown below:



17. Select Old Lookup activity
    a. Go to settings
    b. Configure the properties:
        i. Source dataset: WatermarkSqlDataset
        ii. Use query: Query
        iii. Query: SELECT * FROM Watermark

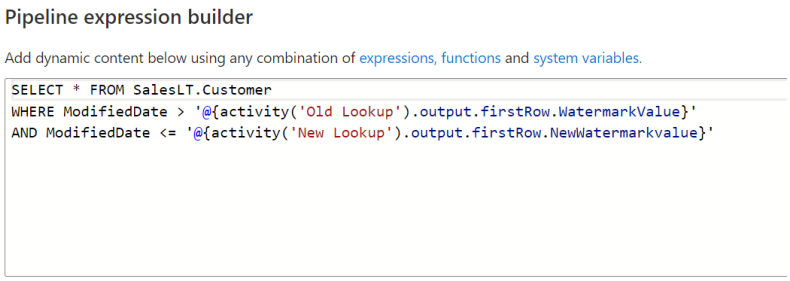


18. Select New Lookup activity
    a. Go to settings
    b. Configure the properties:
        i. Source dataset: CustomerSqlDataset
        ii. Use query: Query
        iii. Query:
        SELECT MAX(ModifiedDate) AS NewWatermarkValue
        FROM SalesLT.Customer

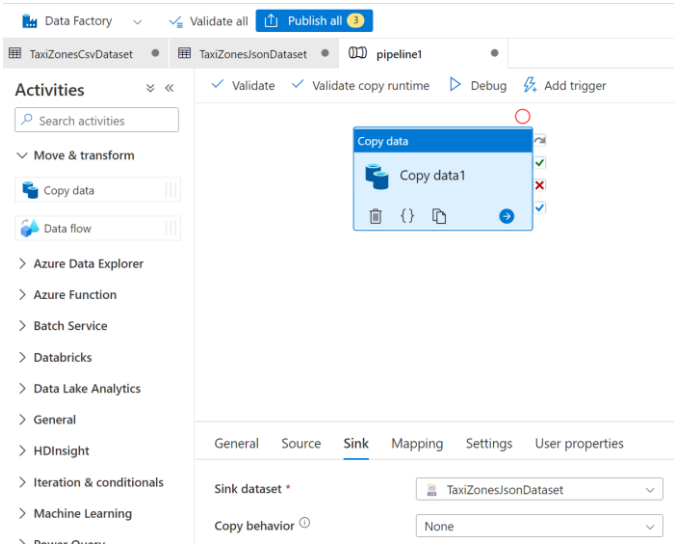


19. Select Copy Data activity
    a. In source tab, select source dataset → CustomerSqlDataset
    b. Use query: Query
    c. Query: (Add dynamic content)

```
SELECT * FROM SalesLT.Customer
WHERE ModifiedDate > '@{activity('Old Lookup').output.firstRow.WatermarkValue}'
AND ModifiedDate <= '@{activity('New Lookup').output.firstRow.NewWatermarkValue}'
```

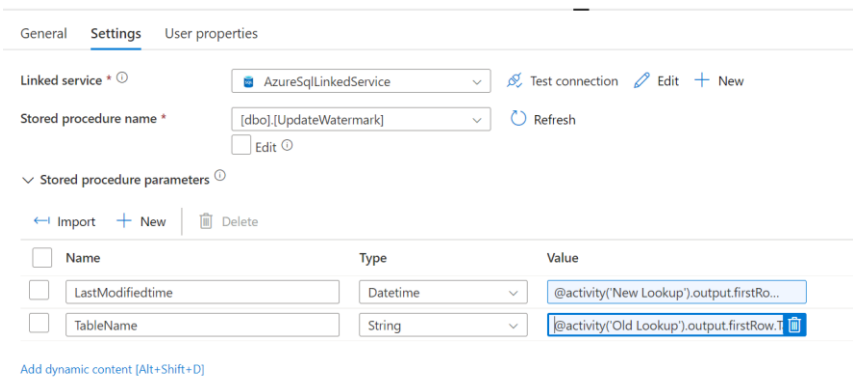


d. In sink tab, select sink dataset →CustomerJsonDataset



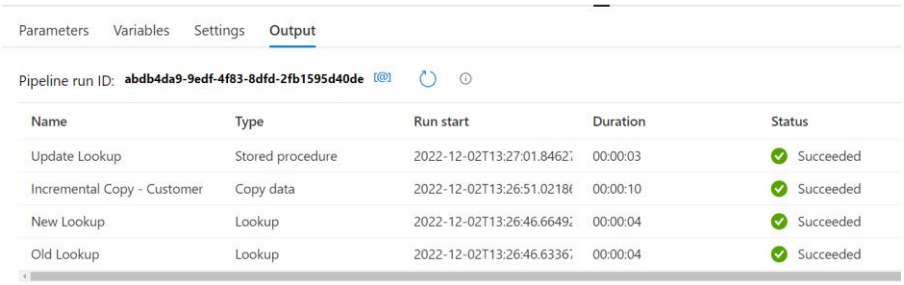
20. Select Update Lookup activity
    a. Go to settings
    b. Configure the properties:
        i. Linked service: AzureSqlLinkedService
        ii. SP Name: UpdateWatermark

    c. Add two stored procedure parameters (as shown in image below):
        i. LastModifiedtime → @activity('Old Lookup').output.firstRow.TableName
        ii. TableName → @activity('Old Lookup').output.firstRow.TableName



21. Once configured, click on Debug to execute the Pipeline

22. Click anywhere in canvas (but not on any activity). Monitor the details



23. Go back to Azure portal → SQL Database → Query Editor

24. Run following command to insert two new records in the table:

```
INSERT INTO SalesLT.Customer (NameStyle, FirstName, LastName, PasswordHash, Password
Salt, rowguid, ModifiedDate)
VALUES ('False', 'Mohit', 'Batra', 'xxx', 'xxx', NEWID(), GETDATE());

INSERT INTO SalesLT.Customer (NameStyle, FirstName, LastName, PasswordHash, Password
Salt, rowguid, ModifiedDate)
VALUES ('False', 'Andrew', 'Smith', 'yyy', 'yyy', NEWID(), GETDATE());
```

25. In Data Factory, debug the pipeline again

26. Monitor in Data factory that only 2 records have moved

27. Check new file in Data Lake and see that it only has 2 records