Azure Data Engineering End to End Project

By Praveen Samudrala

Contents

Problem Statement: Migrate On-premise PostgreSQL database to Azure cloud Data Warehouse from where the structured data can be used for Analytics using PowerBI.

Tables in PostgreSQL to migrate

Query | Query History

```
1  select schemaname, tablename from pg_tables
2  where schemaname = 'public'
```

Data Output | Messages | Notifications

	schemaname name	tablename name
1	public	departments
2	public	employees
3	public	regions
4	public	fruit_imports
5	public	fruits

Table Schemas:

Query | Query History

```
1  select table_schema, table_name, column_name, is_nullable, data_type, character_maximum_length from information_schema
2  where table_name = 'departments'
3
4
```

Data Output | Messages | Notifications

	table_schema name	table_name name	column_name name	is_nullable character varying (3)	data_type character varying	character_maximum_length integer
1	public	departments	department	NO	character varying	100
2	public	departments	division	YES	character varying	100

Query | Query History

```
1  select table_schema, table_name, column_name, is_nullable, data_type, character_maximum_length from information_schema.
2  where table_name = 'employees'
3
4
```

Data Output | Messages | Notifications

	table_schema name	table_name name	column_name name	is_nullable character varying (3)	data_type character varying	character_maximum_length integer
1	public	employees	region_id	YES	integer	[null]
2	public	employees	hire_date	YES	date	[null]
3	public	employees	salary	YES	integer	[null]
4	public	employees	employee_id	NO	integer	[null]
5	public	employees	gender	YES	character varying	1
6	public	employees	department	YES	character varying	17
7	public	employees	first_name	YES	character varying	50
8	public	employees	last_name	YES	character varying	50
9	public	employees	email	YES	character varying	50

Query | Query History

```
1  select table_schema, table_name, column_name, is_nullable, data_type, character_maximum_length from information_schema.
2  where table_name = 'regions'
3
4
```

Data Output | Messages | Notifications

	table_schema name	table_name name	column_name name	is_nullable character varying (3)	data_type character varying	character_maximum_length integer
1	public	regions	region_id	NO	integer	[null]
2	public	regions	region	YES	character varying	20
3	public	regions	country	YES	character varying	20

No limit

Query | Query History

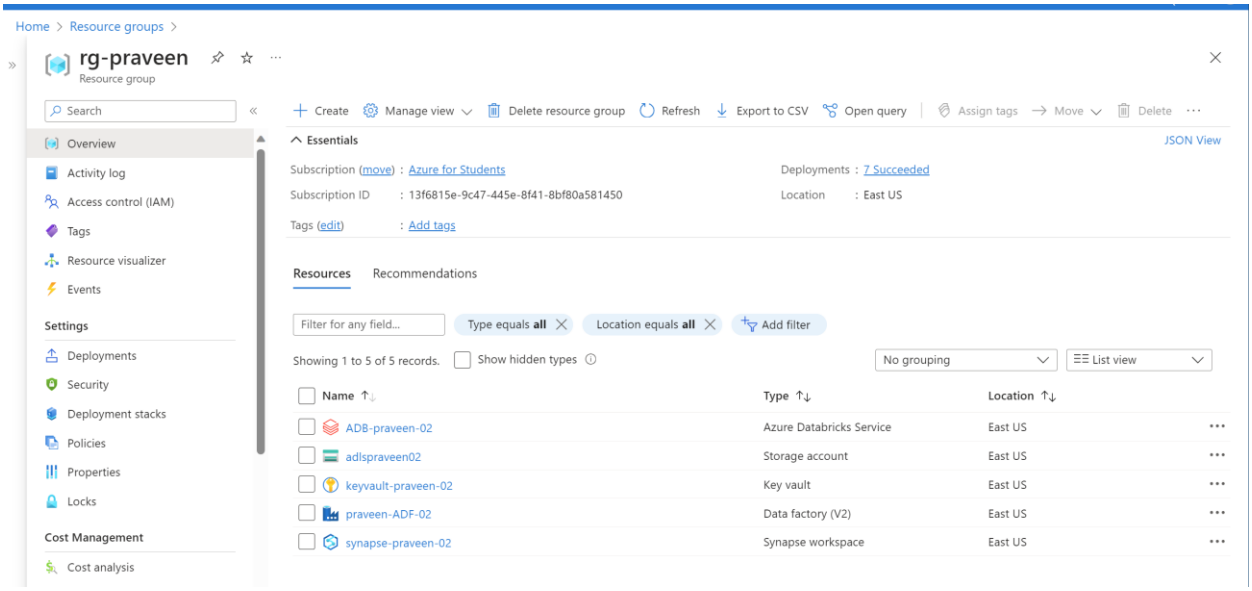
```
1  select table_schema, table_name, column_name, is_nullable, data_type, character_maximum_length from information_schema.
2  where table_name = 'fruit_imports'
3
4
```

Data Output | Messages | Notifications

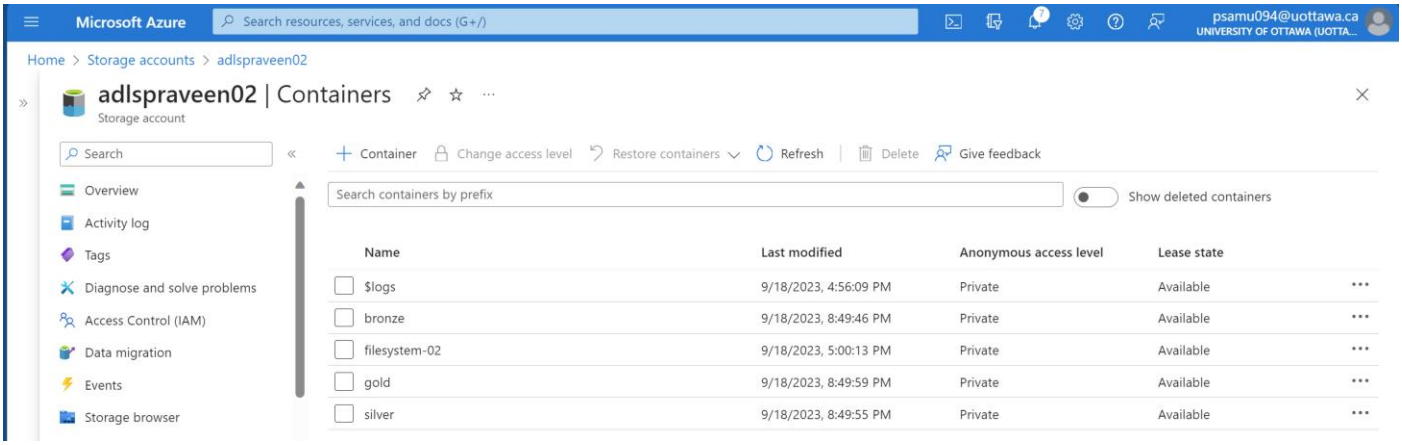
	table_schema name	table_name name	column_name name	is_nullable character varying (3)	data_type character varying	character_maximum_length integer
1	public	fruit_imports	id	YES	integer	[null]
2	public	fruit_imports	supply	YES	integer	[null]
3	public	fruit_imports	cost_per_unit	YES	numeric	[null]
4	public	fruit_imports	name	YES	character varying	20
5	public	fruit_imports	season	YES	character varying	10
6	public	fruit_imports	state	YES	character varying	20

Required Resources:

1. Azure Data Factory
2. Azure Databricks
3. Azure Synapse
4. Azure Data Lake Gen 2
5. Azure Key Vault

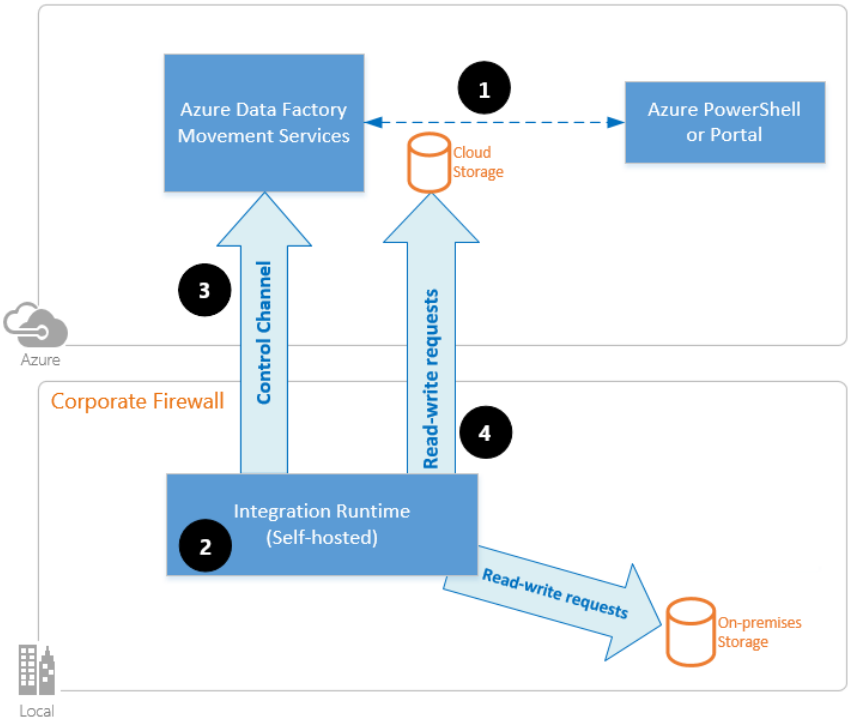


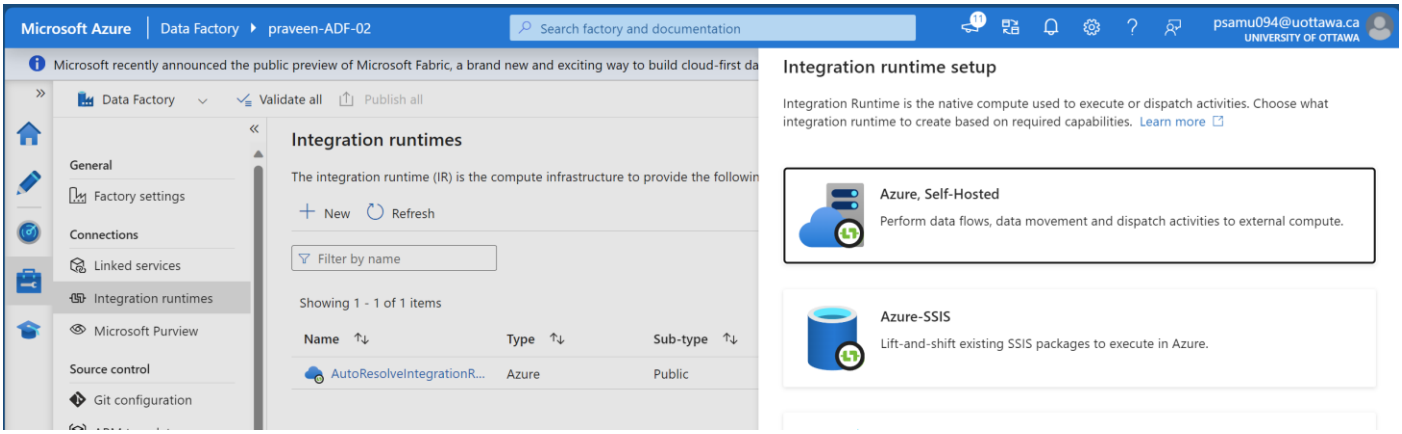
Create Bronze, Silver and Gold containers in ADLS

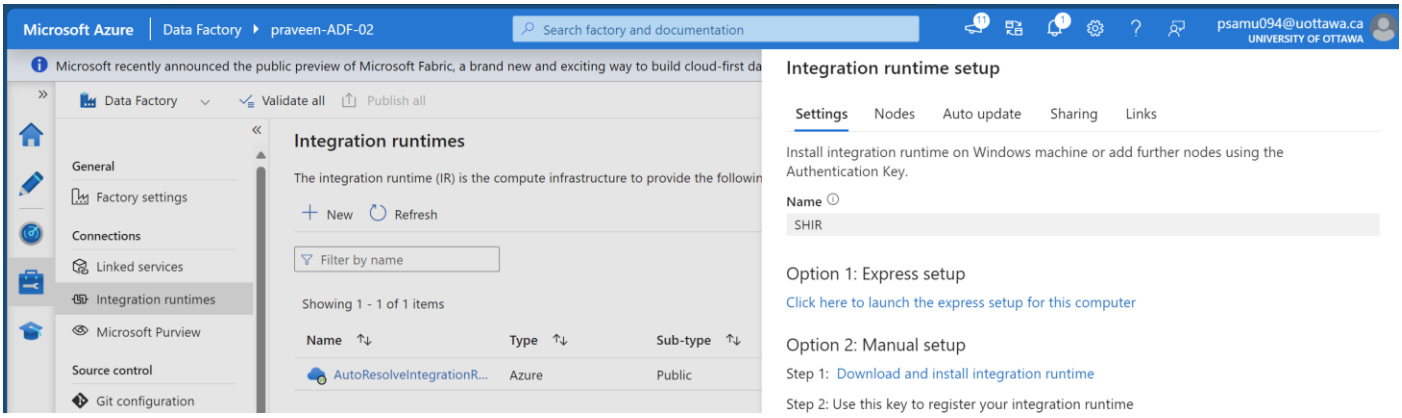


Step 1 – Connecting On-premise PostgreSQL database to Azure using Self-Hosted Runtime.

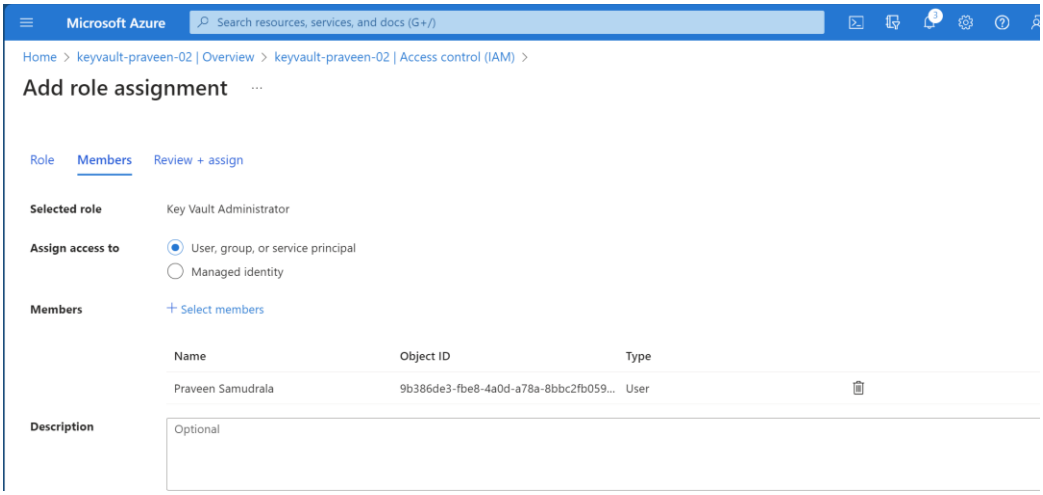
1.1 Create a new Integration Runtime - Self hosted, Install Azure Self Hosted Runtime



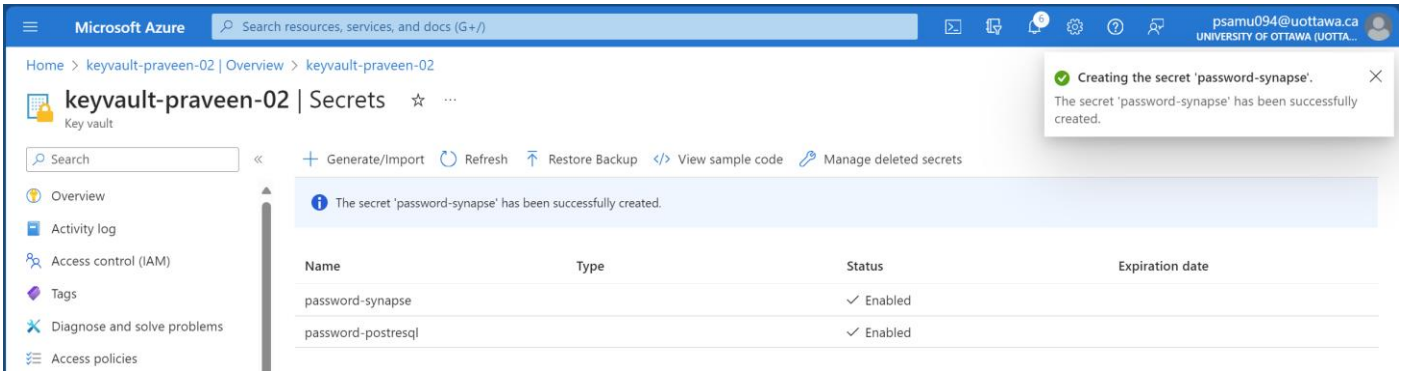




1.2 Give user permission to Key Vault resource

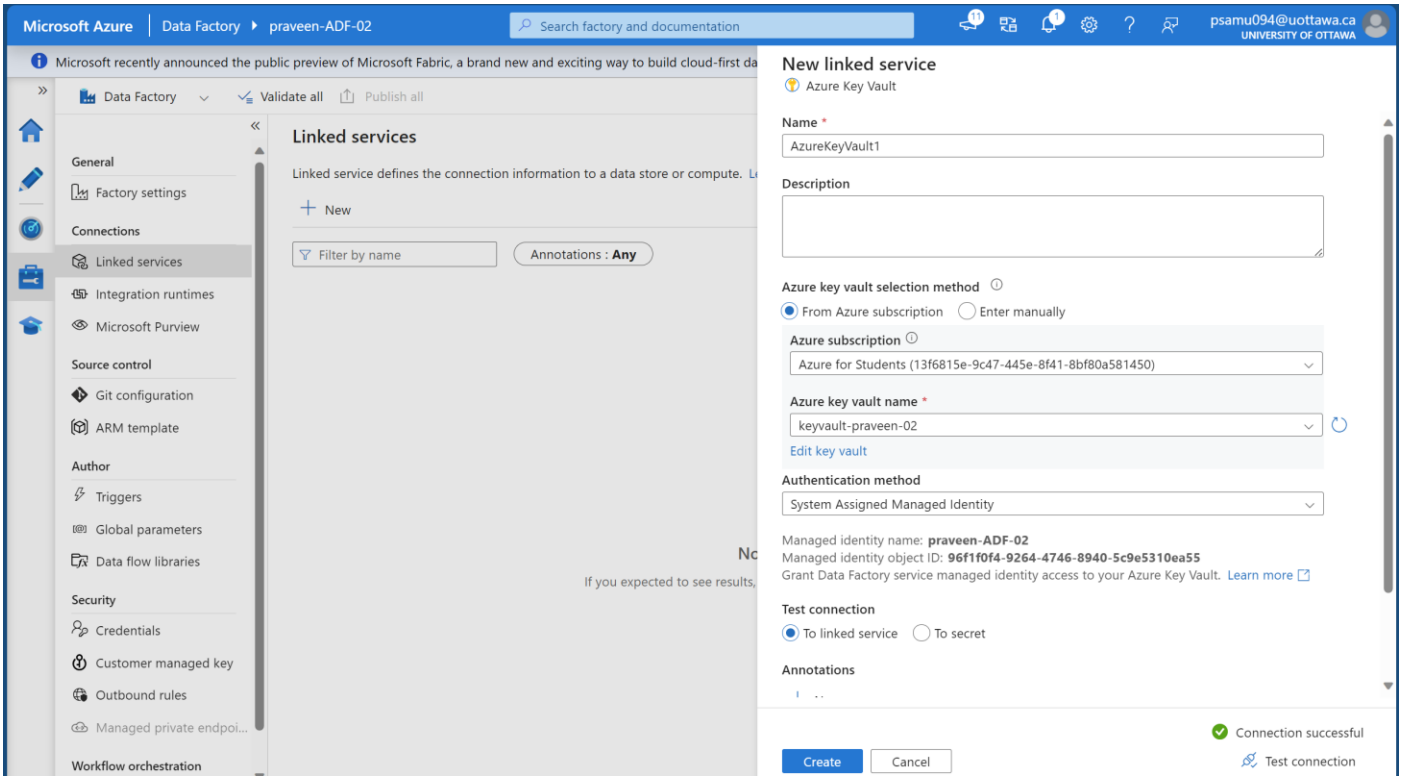


1.3 Setup PostgreSQL username password as Secrets in Azure Key Vault

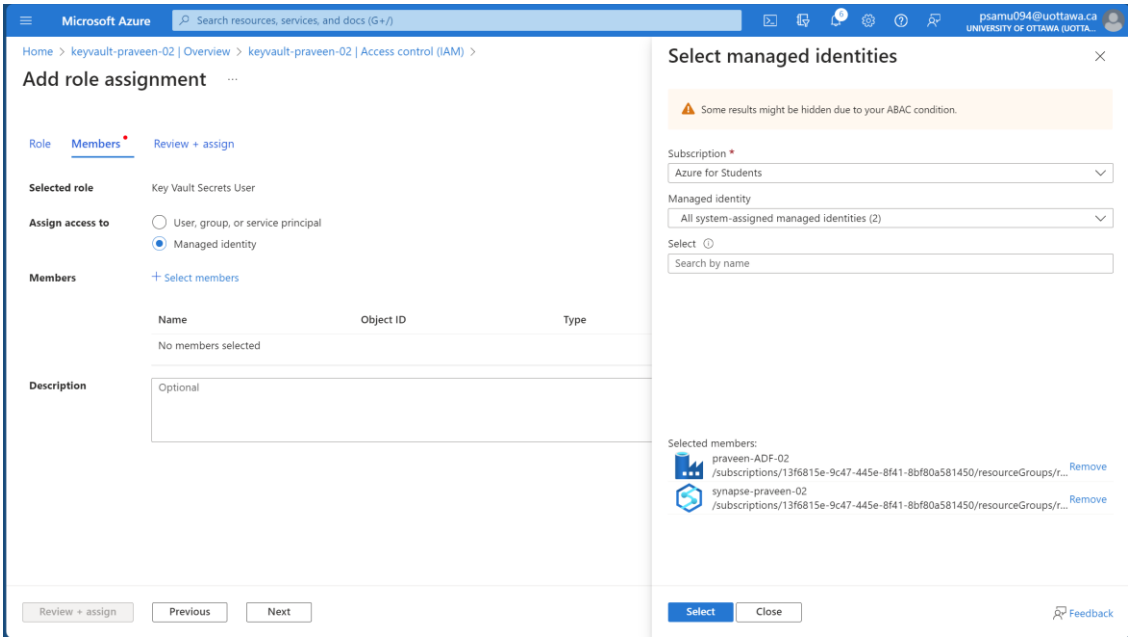


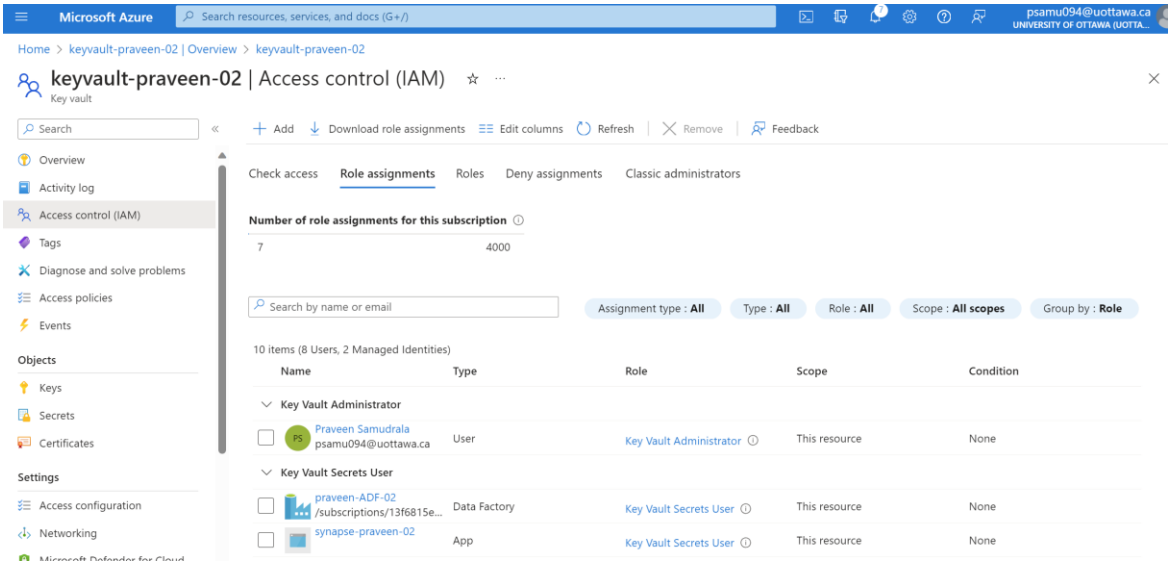
Step 2 – Using Data Factory for Ingestion and loading the data to ADLS.

2.1 Create Linked Service from ADF to Key Vault to access passwords

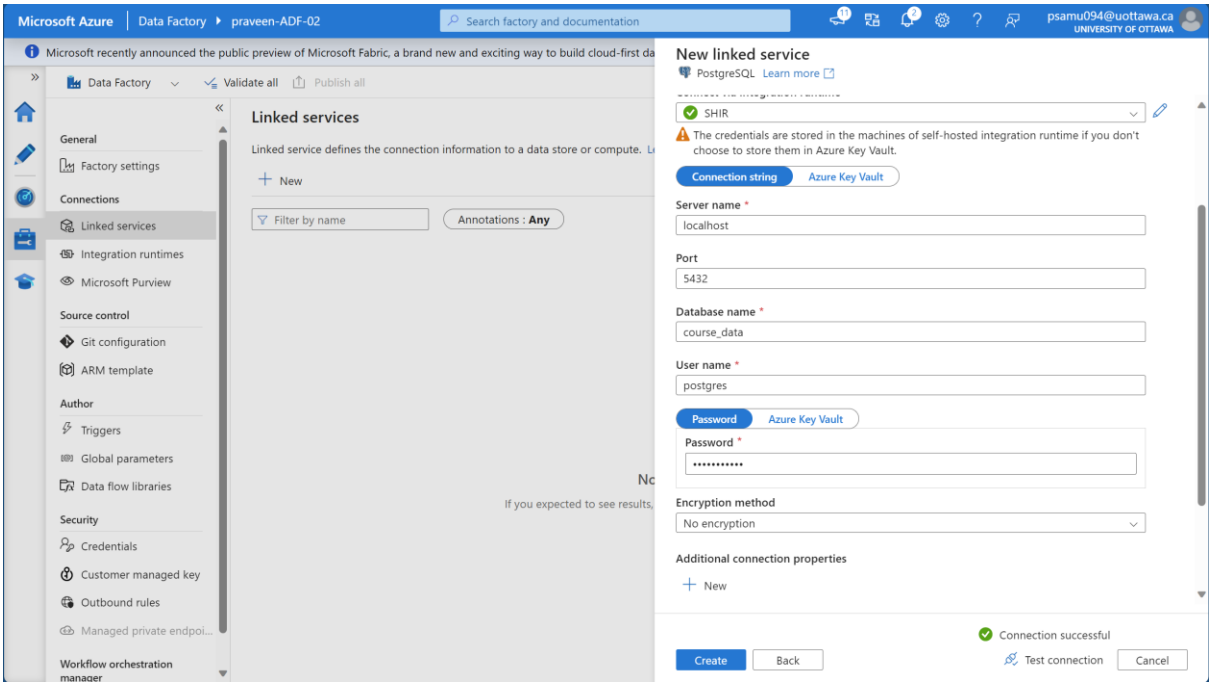


2.2 Allow ADF and Synapse to access Key Vault Secrets





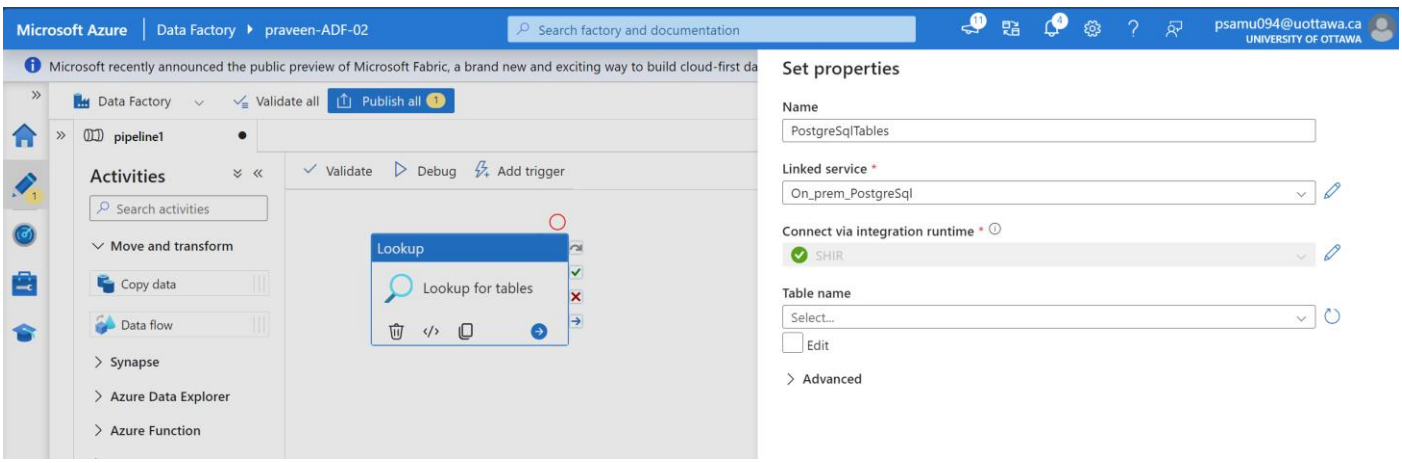
2.3 Create Linked Service from ADF to on-premise Data Factory

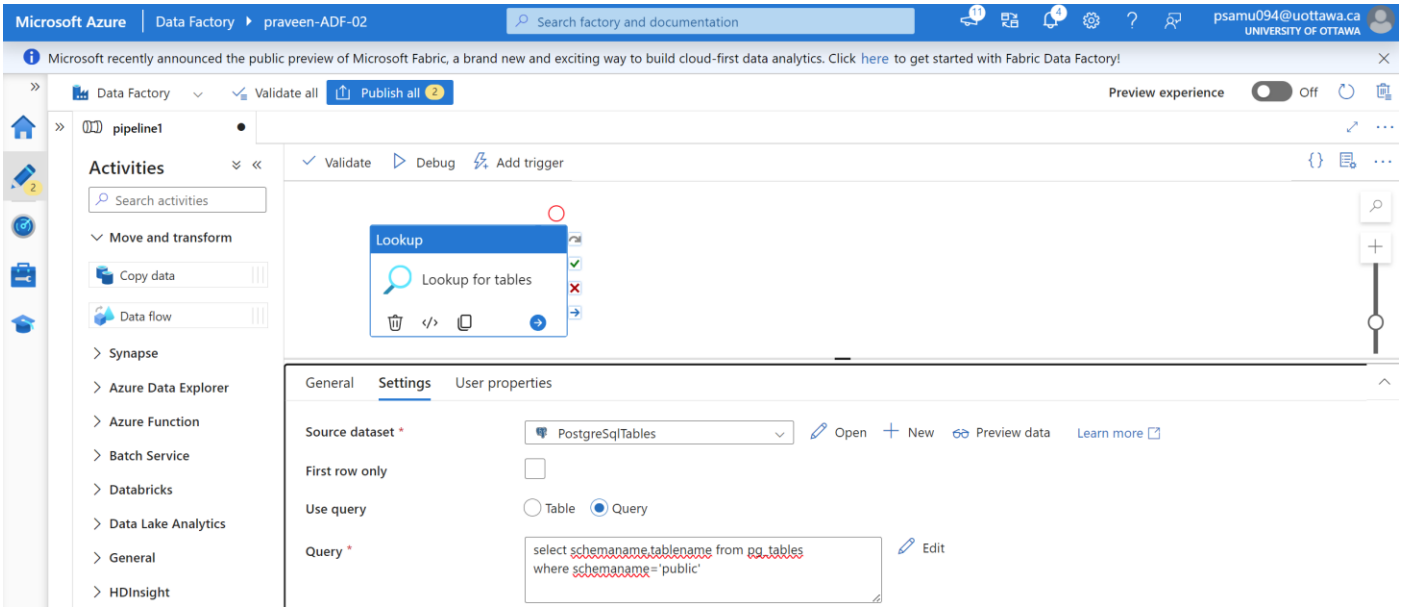


Step 3 – Using Azure Databricks for data transformation and cleaning.

3.1 – Create a Pipeline starting with 'Lookup' activity to look for all tables satisfying a custom query.

Create a Source Dataset

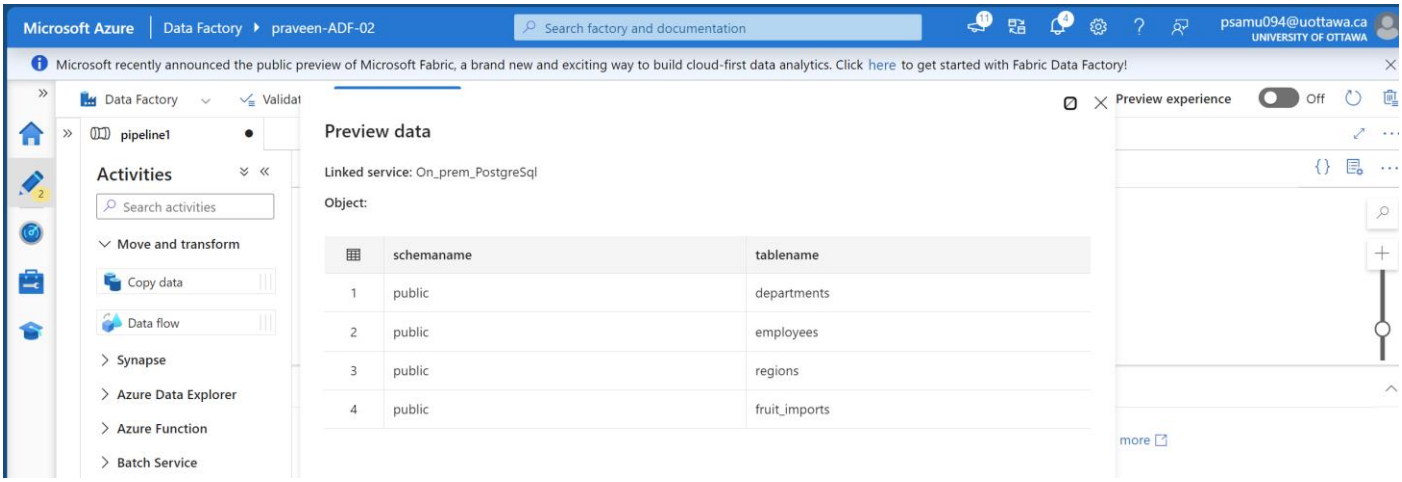




Custom Query:

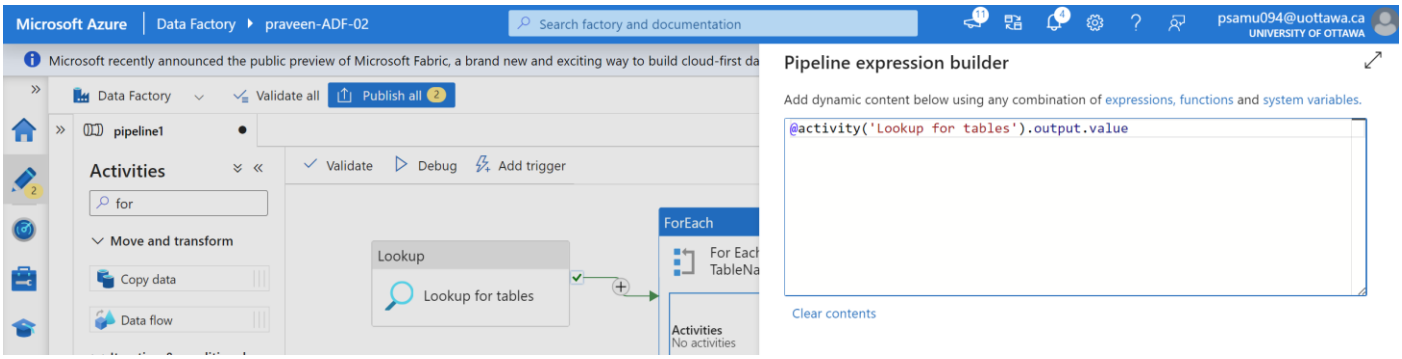
Select schemaname, tablename from pg_tables where schemaname='public'

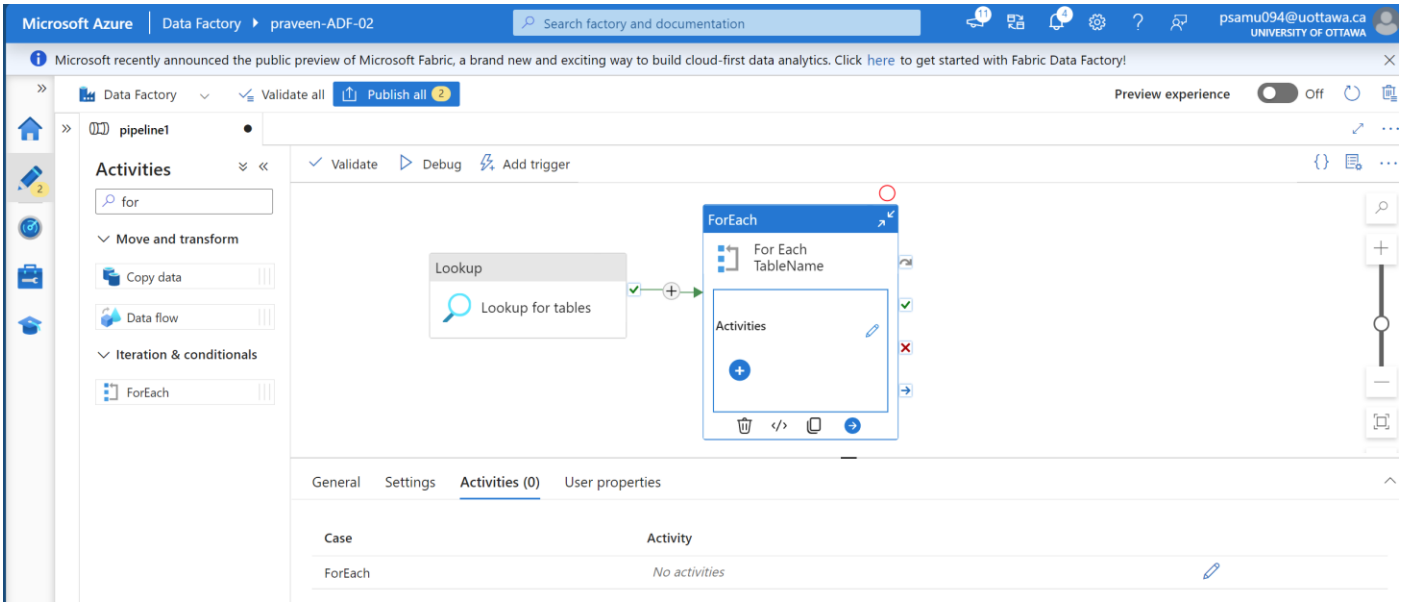
Fetched all tables from PostgreSQL DB



Result from Lookup activity result in a JSON object telling the count and values of the result. In our case, the count is 4, values is an array of scshename,tablename.

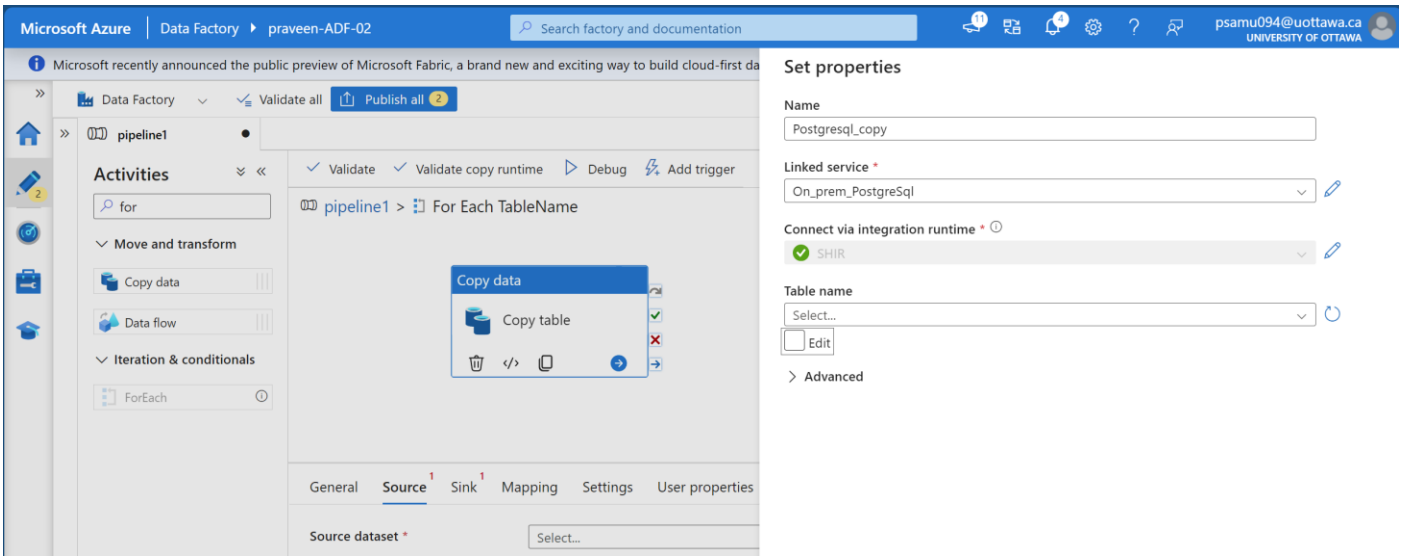
3.2 Add 'ForEach' activity after Lookup activity and add the dynamic expression

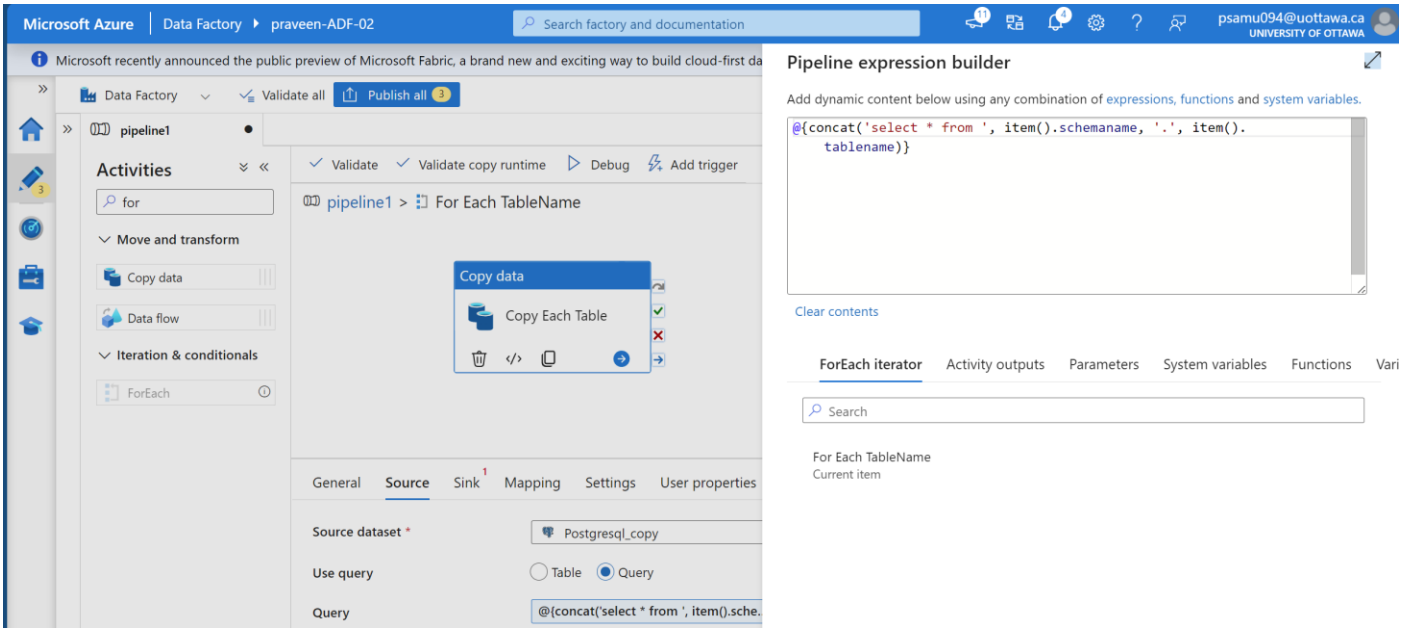




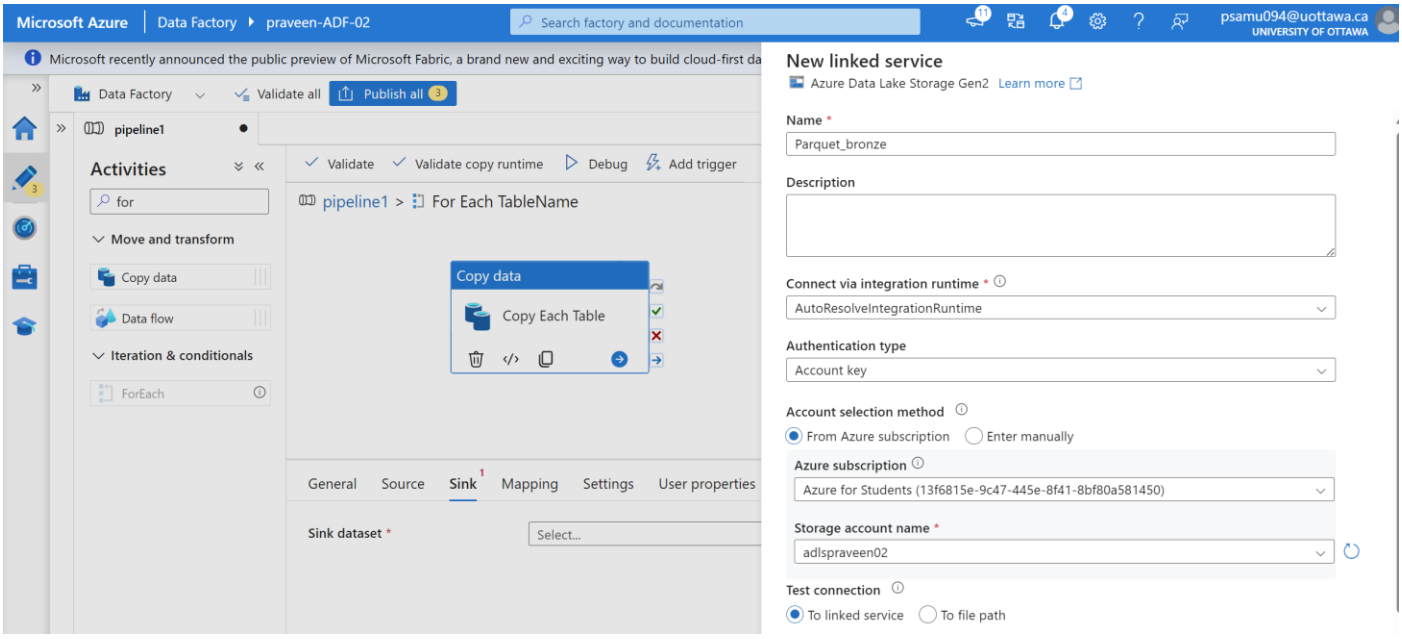
Configure the Activity – Lookup activity returns the table names, for each table, 'ForEach' activity need to fetch all data from each table.

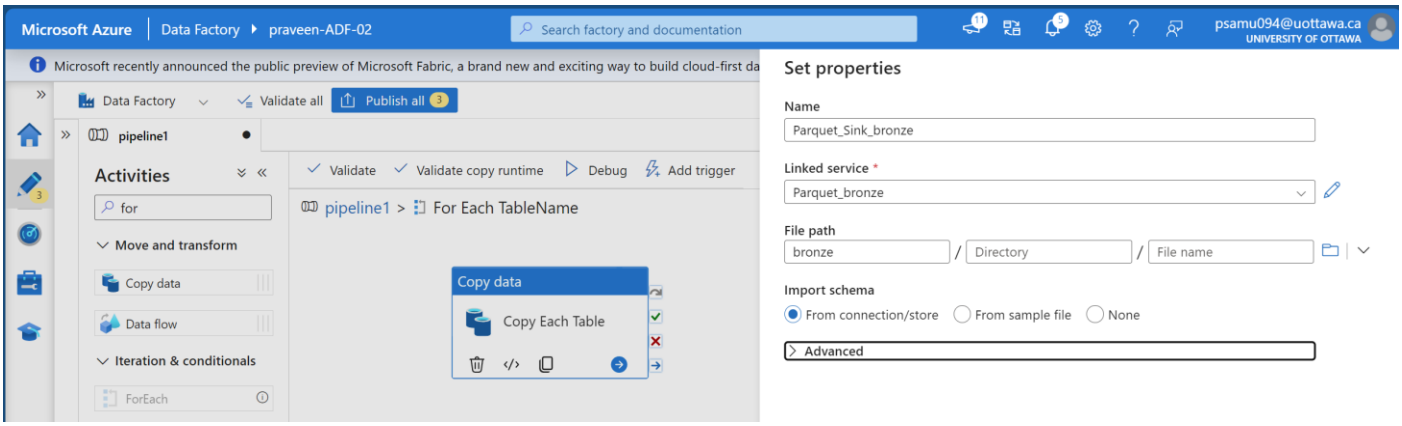
Edit the ForEach activity, then setup a Copy activity





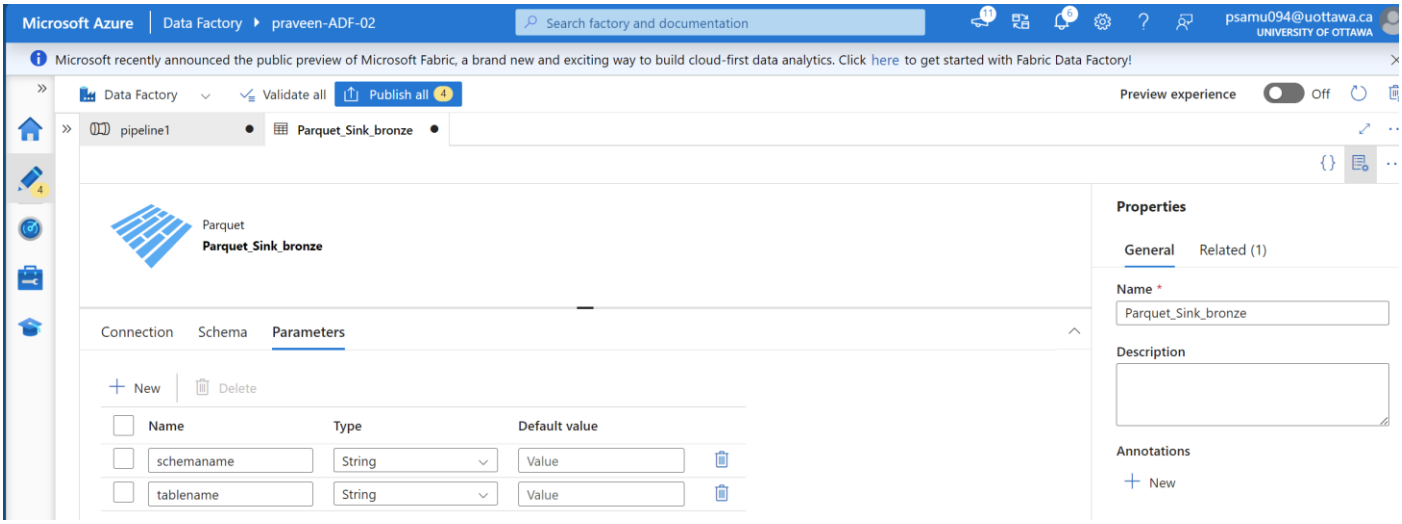
Create a Sink – ADLS Gen 2 with Parquet Format. Create a new Linked service to ADLS, then create a Sink Dataset.



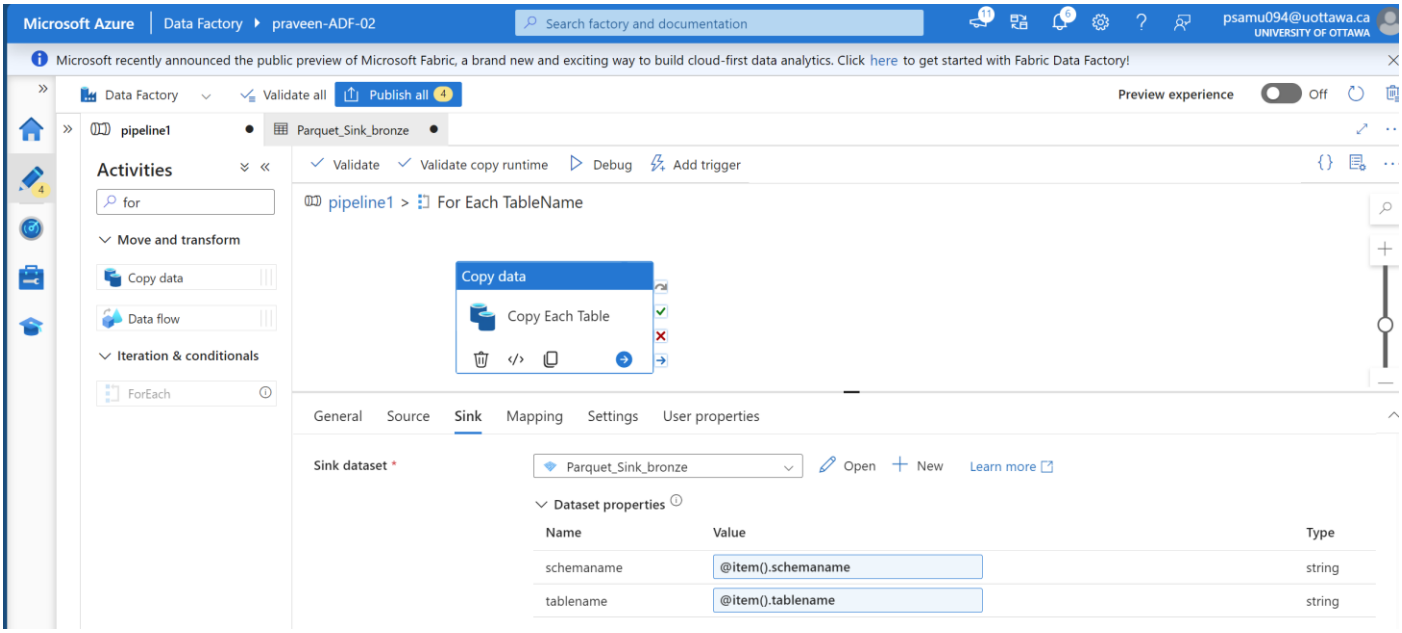


I want the folder structure inside bronze container as bronze/schemaname/tablename/table.parquet

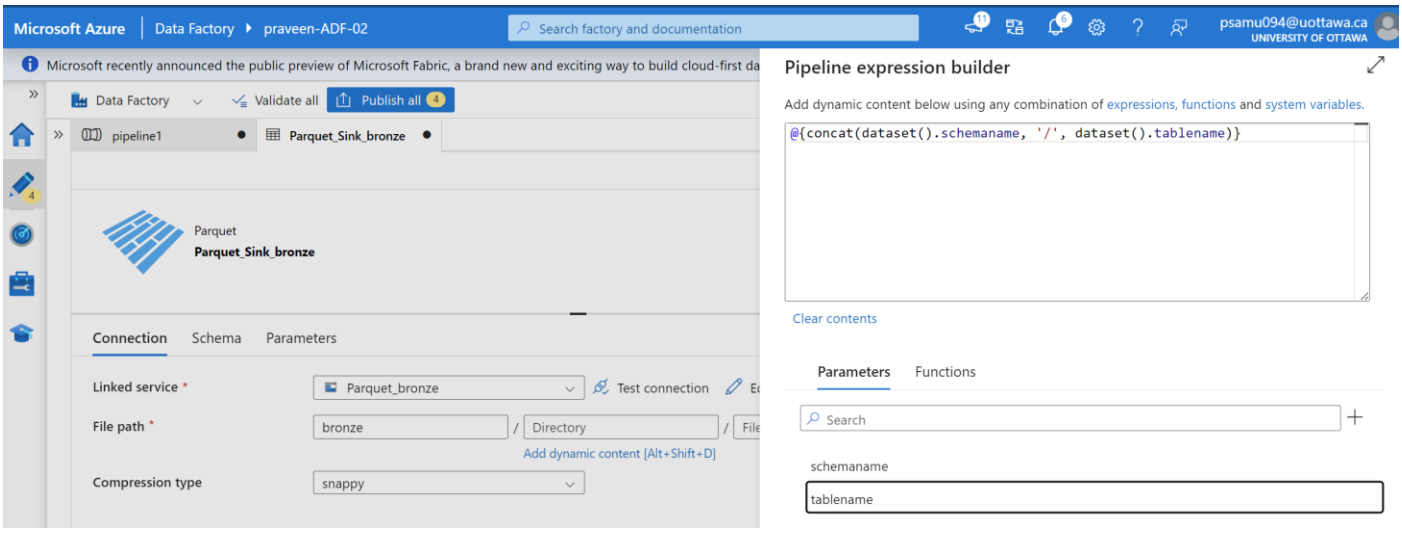
Create parameters in the sink dataset so that they can be used for folder structure

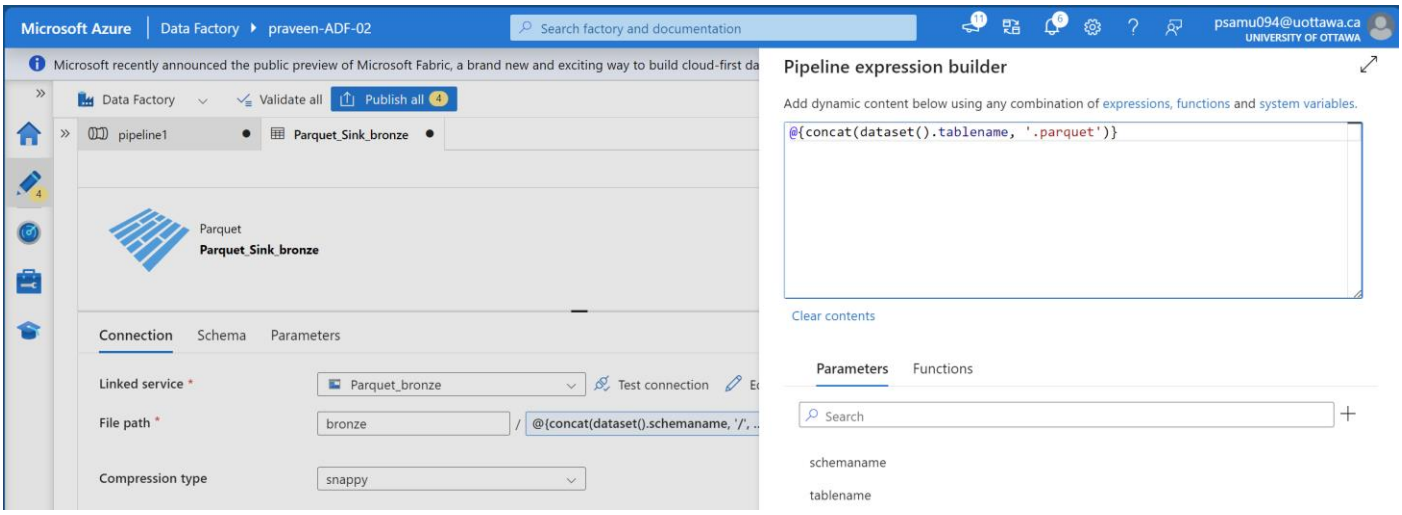


Assign dynamic values as per the schema and table names that we get from ‘LookUp’ activity.

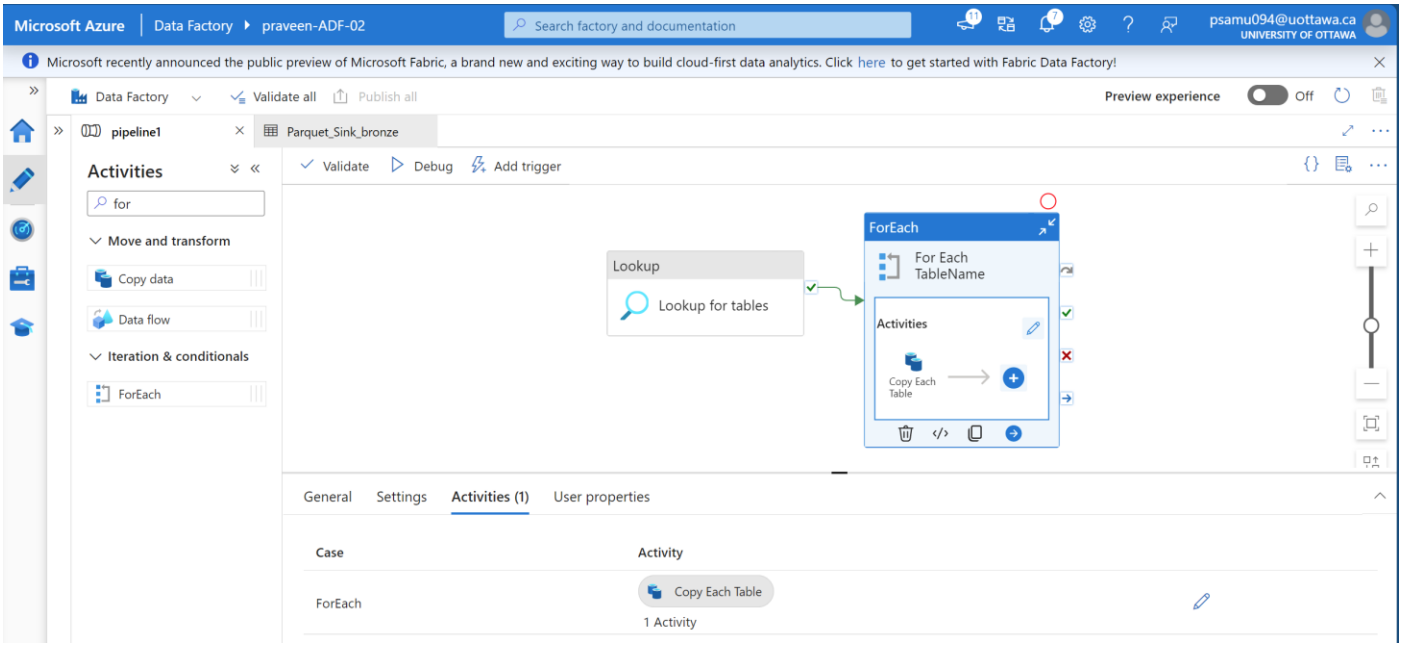


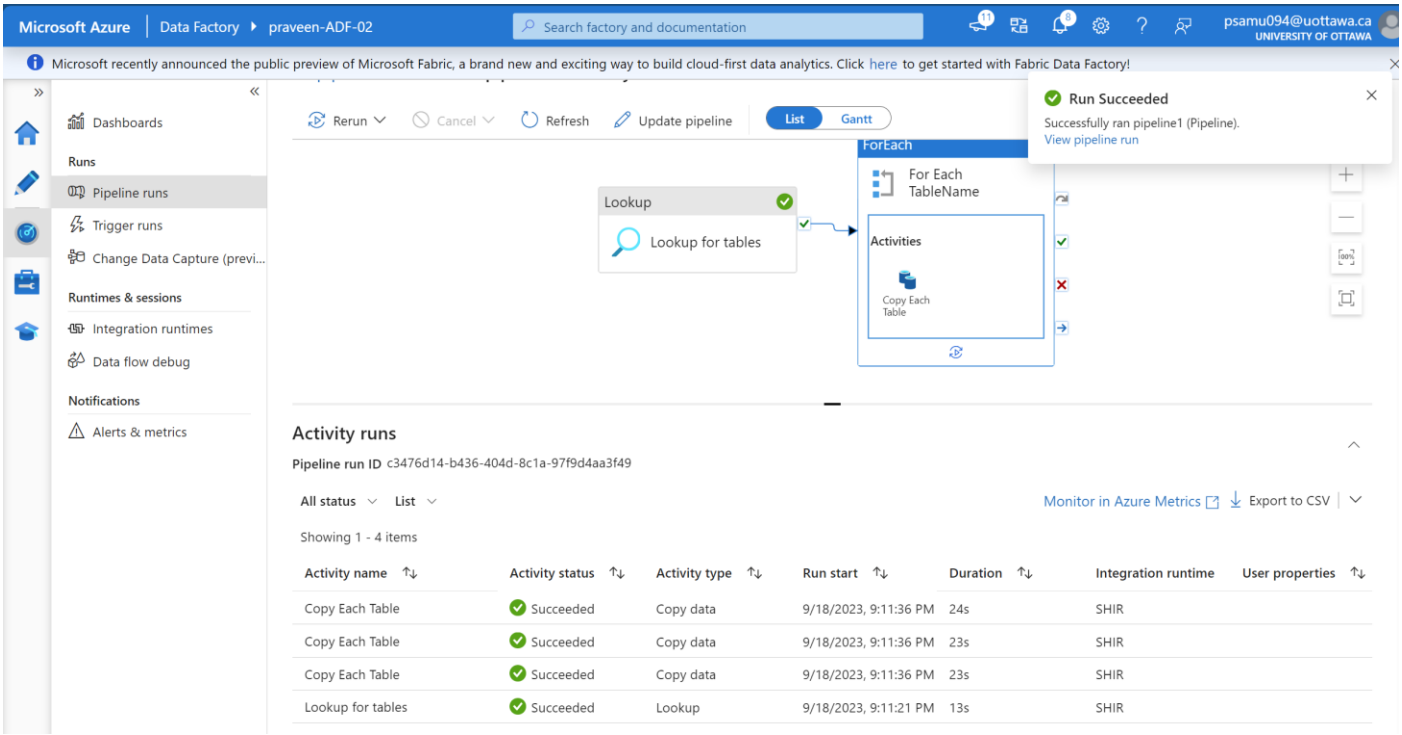
Create the folder structure



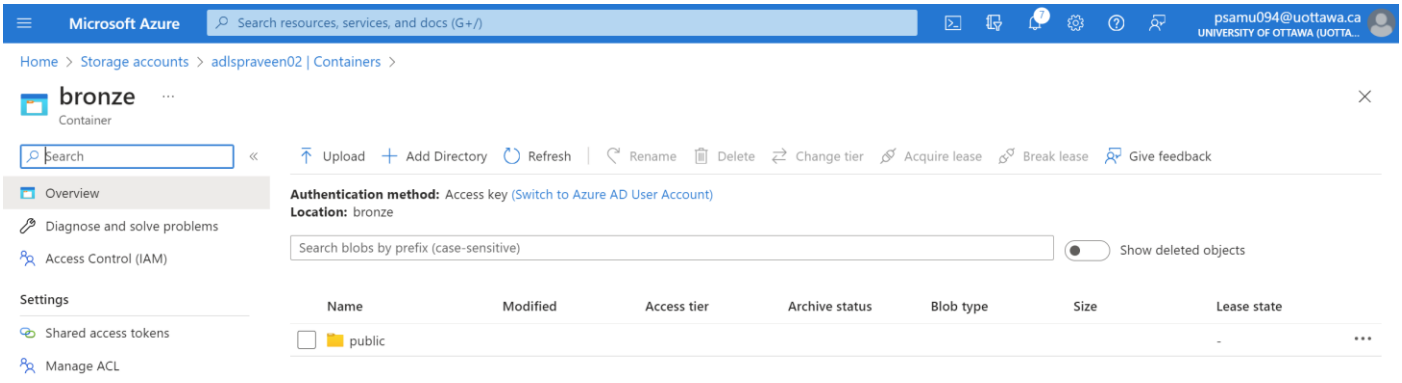


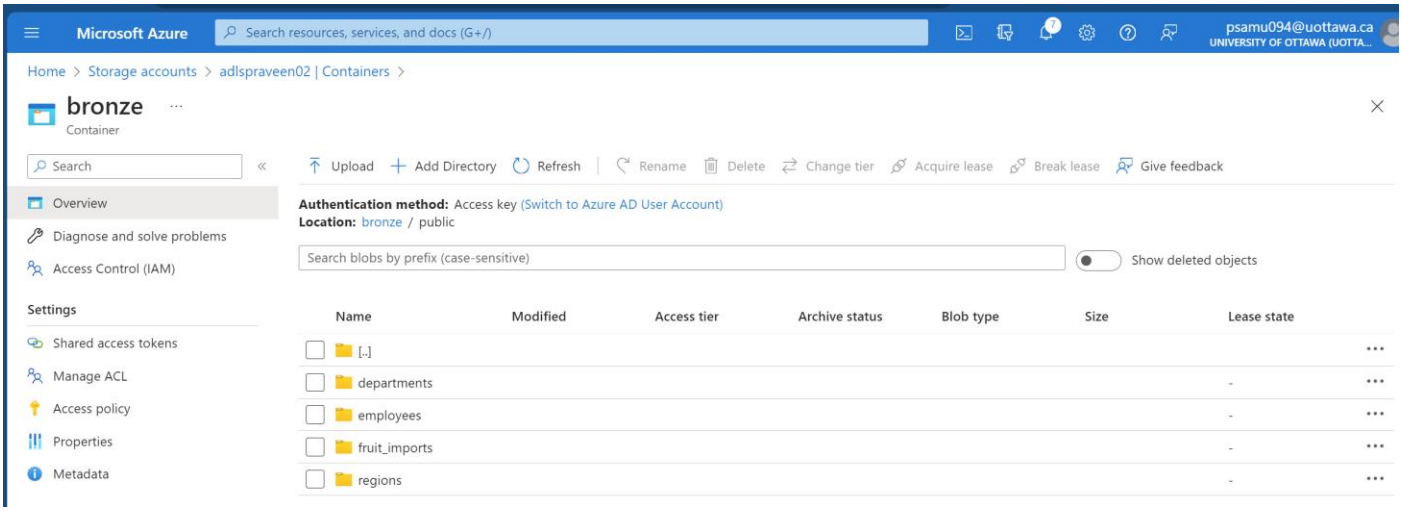
Publish all the files and Execute the Pipeline to check if tables are fetched and stored in ADLS in Parquet format in the desired folder structure.

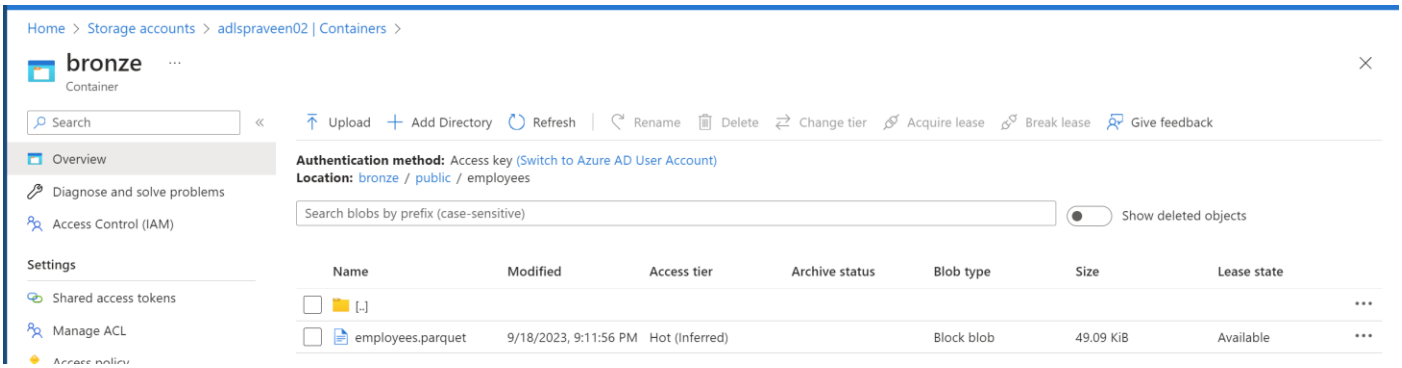


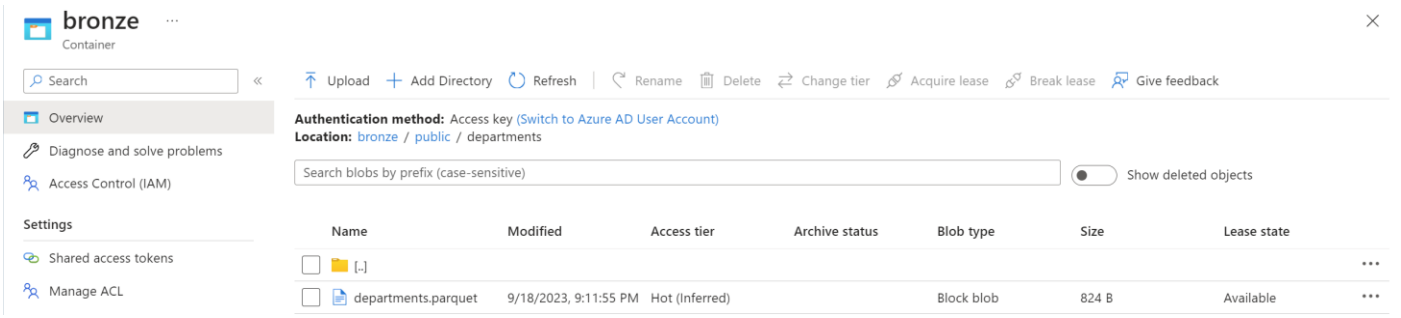


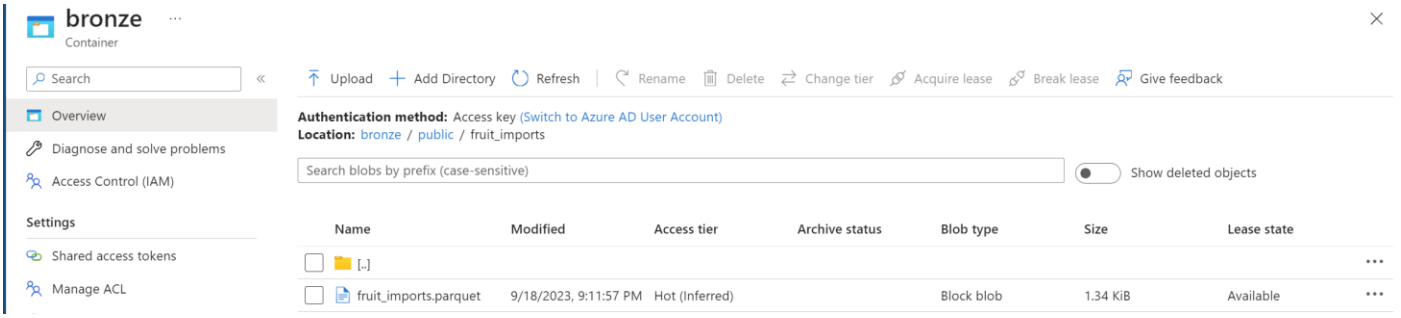
Verifying files in ADLS











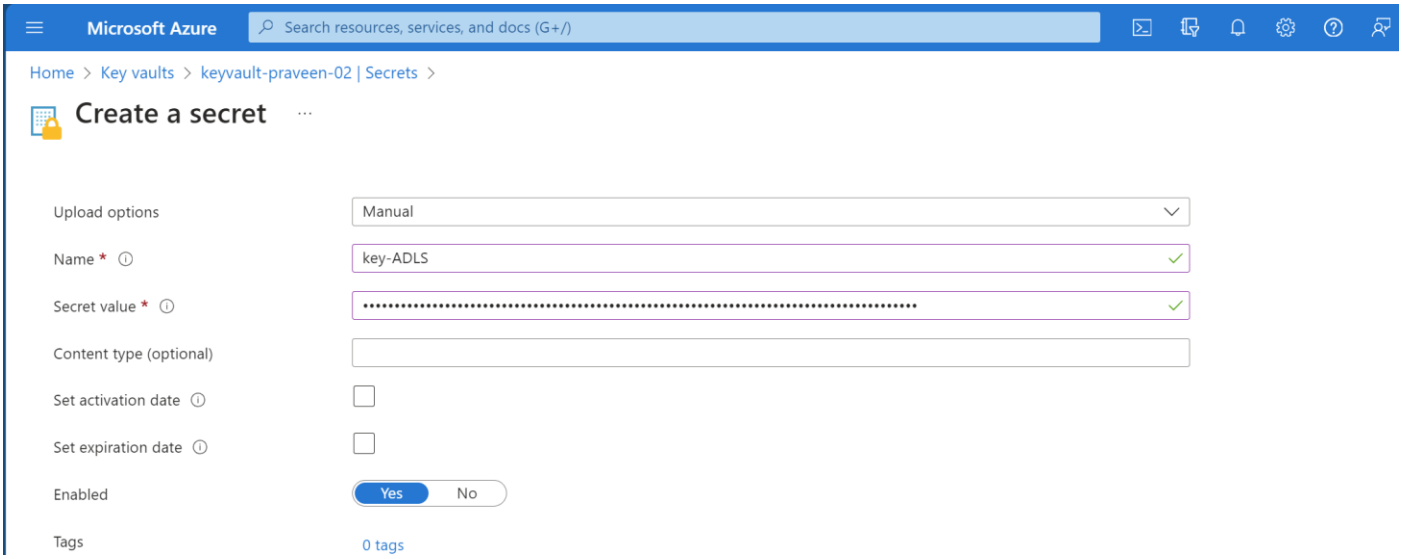
Step 4 – Data Transformation in Azure Databricks

4.1 – Developing in Azure Databricks

Create Azure Databricks Compute Cluster and Notebook – Single Compute and Unrestricted.

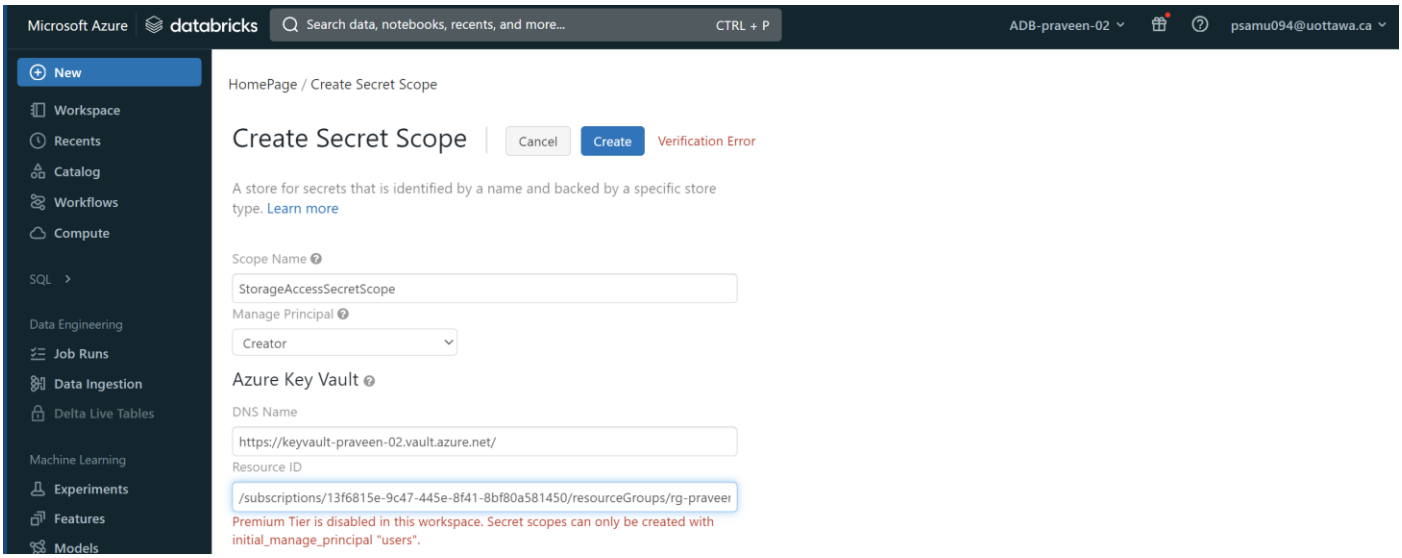
Give access to Databricks cluster and notebook to access ADLS storage. As Premium Tier is disabled in my workspace, neither can I enable 'Credential Passthrough' nor I can create a secretScope to use Azure Key Vault to access the ADLS access key, hence giving the access key in the spark configuration manually.

If Premium Tier is enabled, either enable 'Credential Passthrough' in cluster settings, or Create a secret in Azure Key Vault with secret as ADLS Access Key



In Databricks workspace, add '#secrets/createScope' to the URL to create a Secret Scope

Give a Scope name, for DNS go to Azure Key Vault -> Choose resource -> Properties -> Vault URI, resource ID -> resource ID.



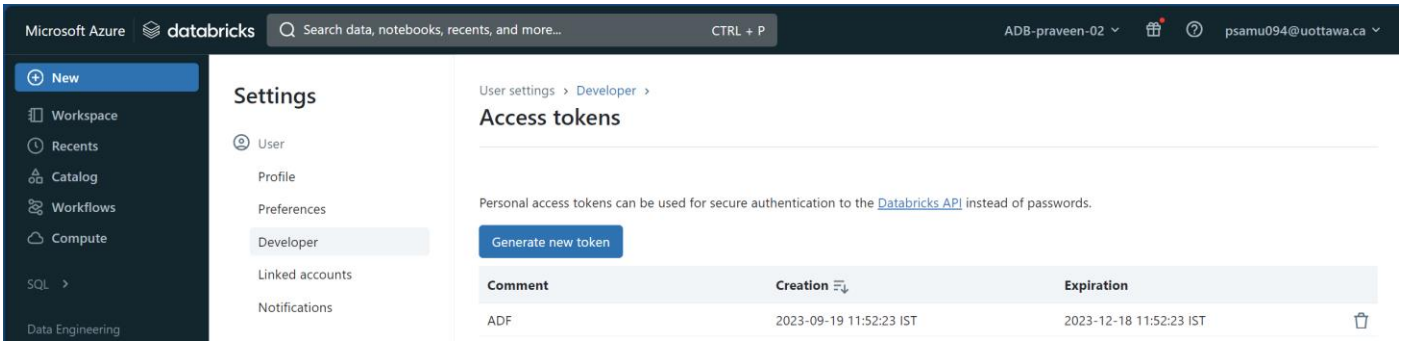
Getting error as I don't have Premium Tier.

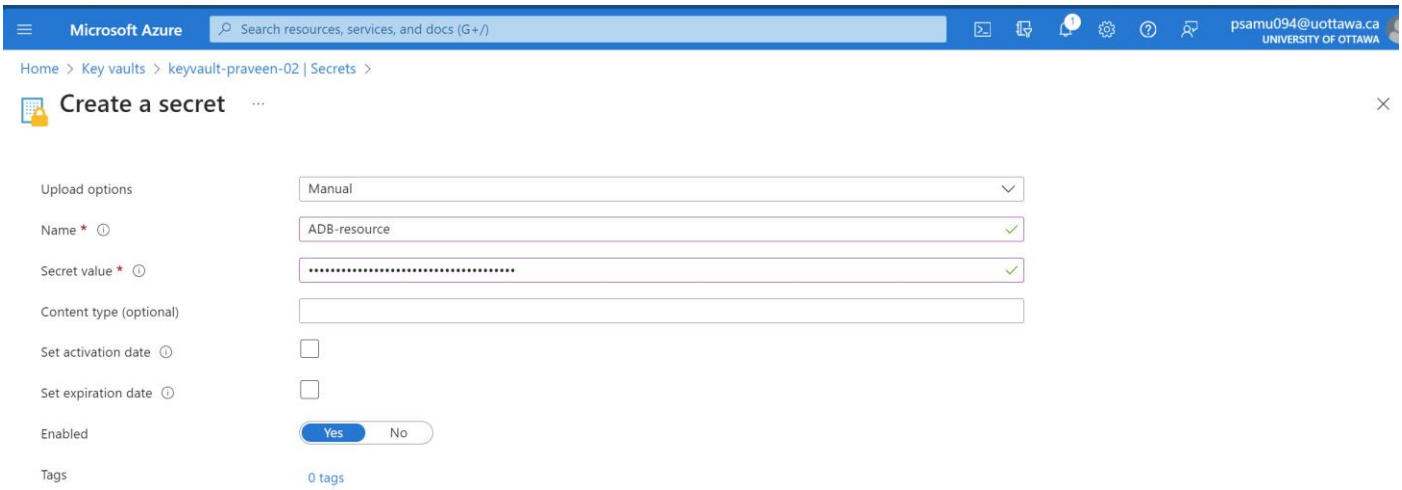
After manually setting ADLS key in Spark Config. Develop the code for transforming data field in tables and save to Silver container. Usually during transformation to Gold stage, tables are brought into fact and dimension tables format. As in our case the tables are already in this format, not performing any transformation from Silve to Gold stage. Reading tables from Silver container and saving to gold container.

4.2 – Making Azure Databricks notebook part of Data Factory Pipeline.

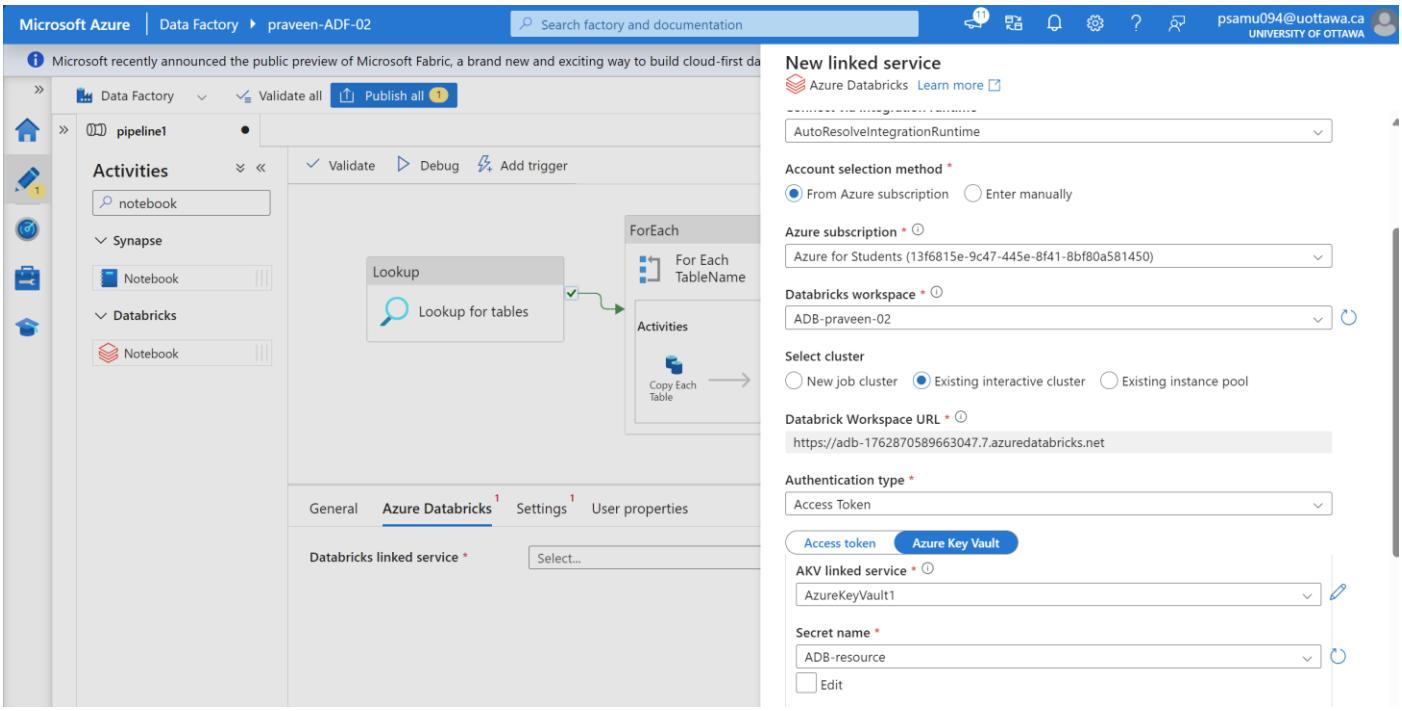
In Azure Databricks workspace -> Name -> User Settings -> Developer -> Access Tokens

Generate a New Token and save it in Azure Key Vault, it will be used in Data Factory Linked Service to Databricks resource.

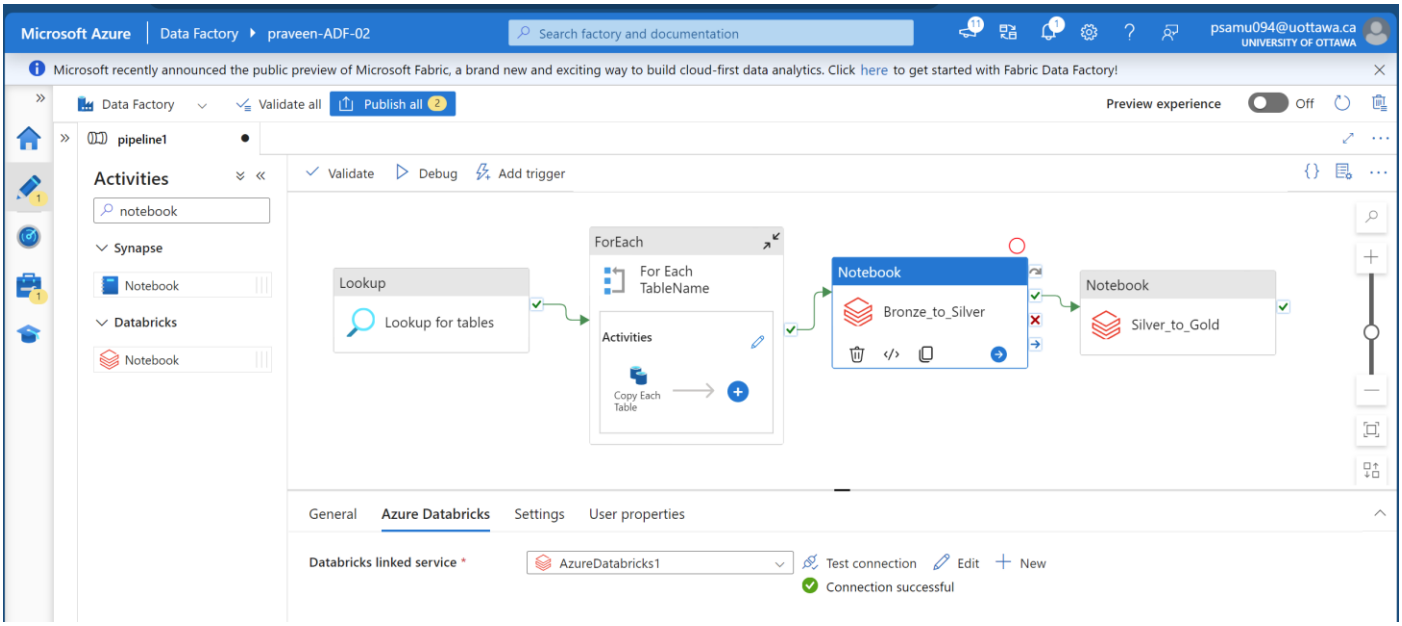




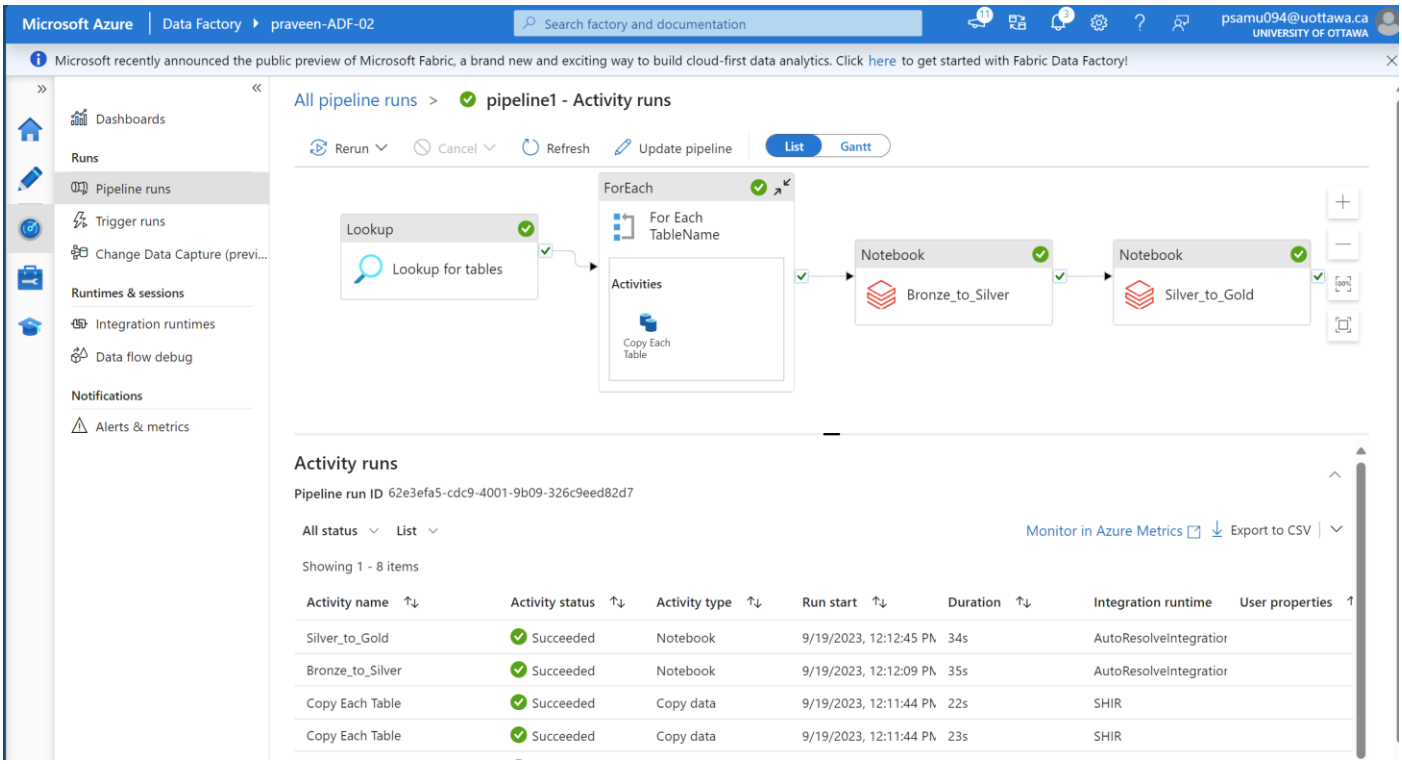
In the Data Factory Pipeline, create a Linked Service to Azure Databricks resource.



Add Azure Databricks Notebooks in the Pipeline each triggering Transformation to Silver stage and Gold stage.

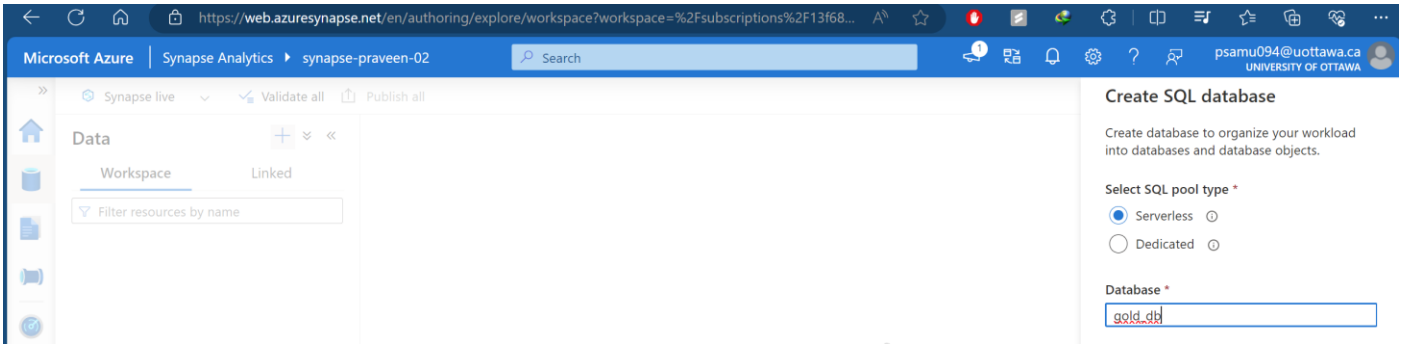


Trigger to execute the pipeline.

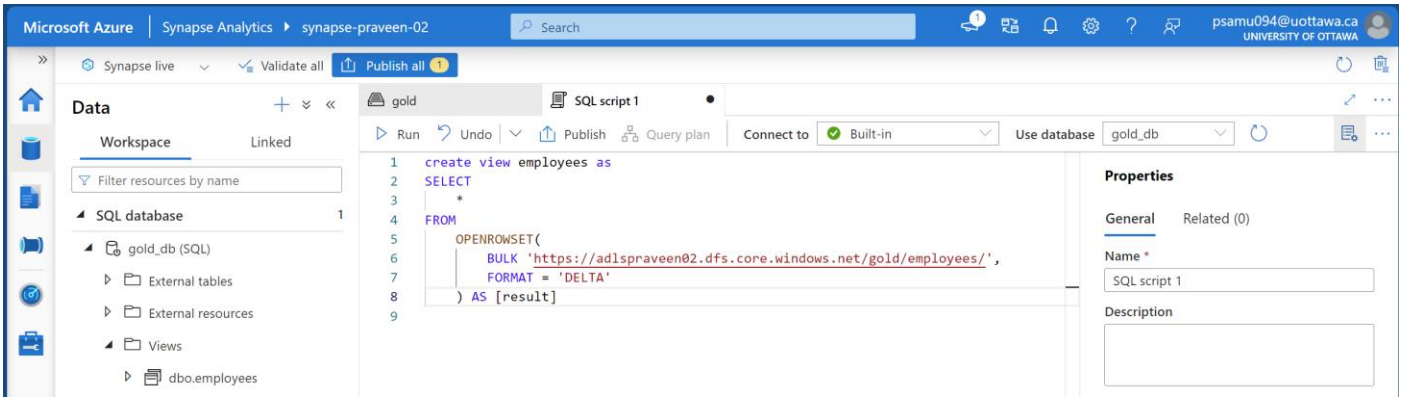


Step 5 – Loading Delta Tables from Databricks to Azure Synapse as Data Warehouse

5.1 Create a Serverless SQL DB in Synapse

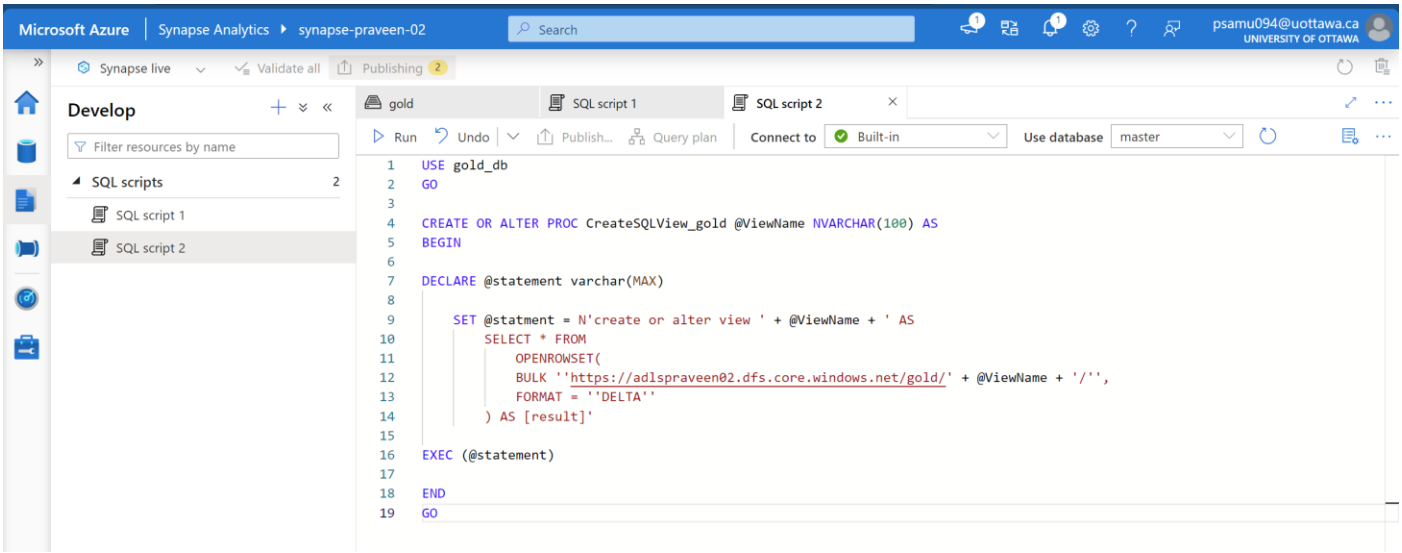


Learn – Views can be created in the SQL DB using SQL in Synapse like below

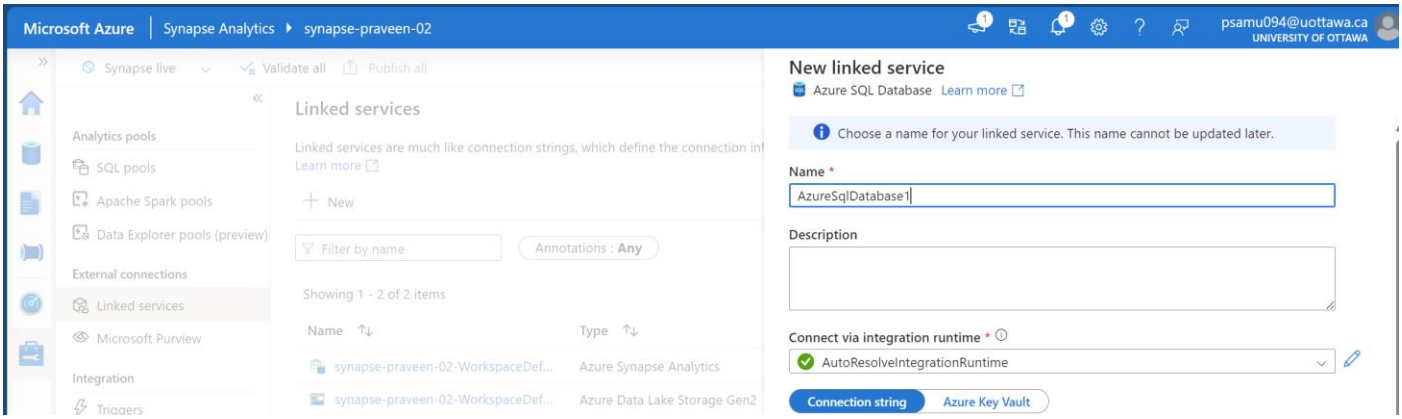


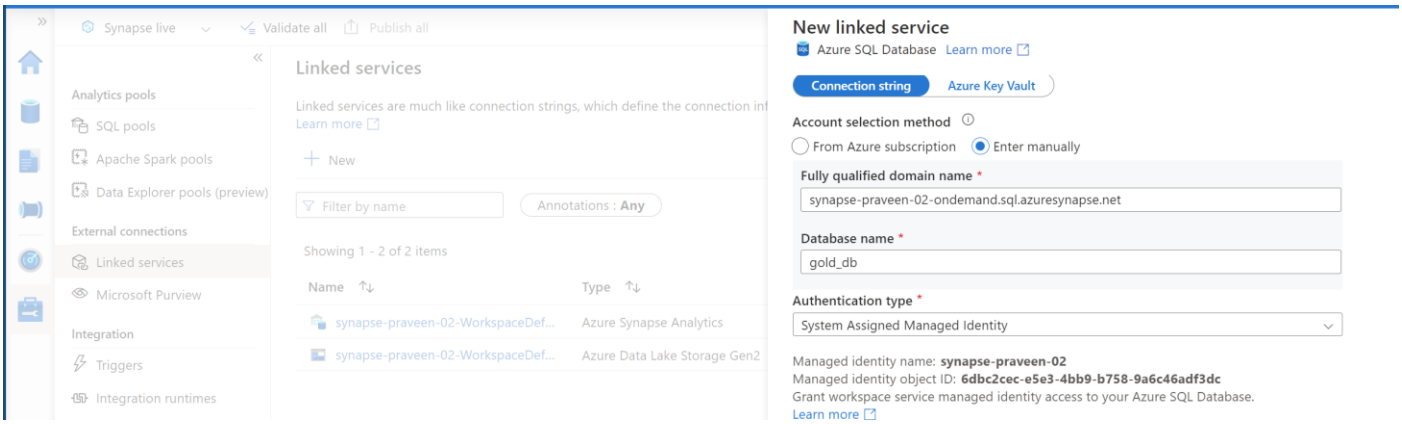
5.2 Now I'll create a Pipeline that can be invoked to create views in SQL DB automatically.

Develop a SQL script such that a View for each table is created dynamically as the tables are read.



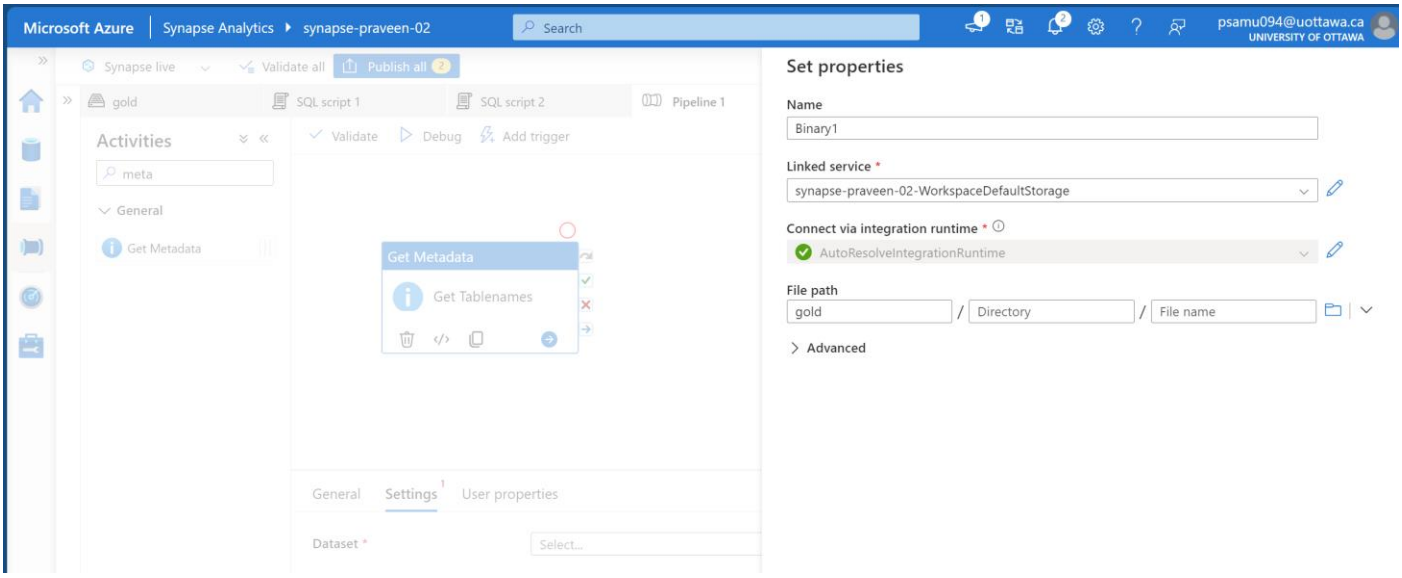
As we are creating the views in SQL DB, Synapse pipeline needs Linked Service to Azure SQL DB to connect and create views.



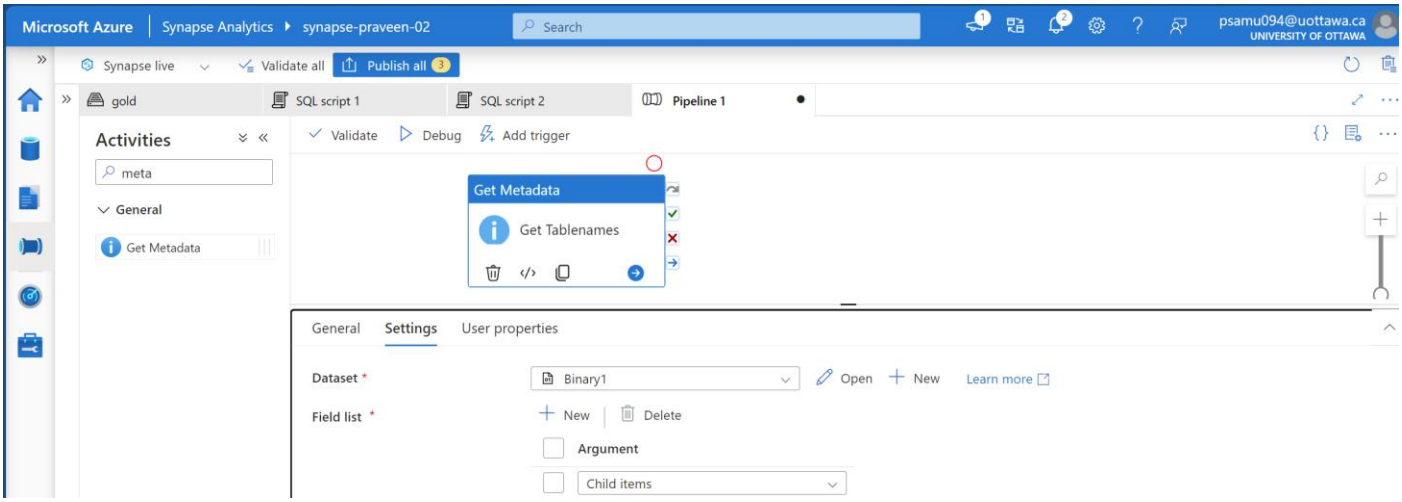


Domain Name = Synapse Workspace -> Properties -> Serverless SQL Endpoint

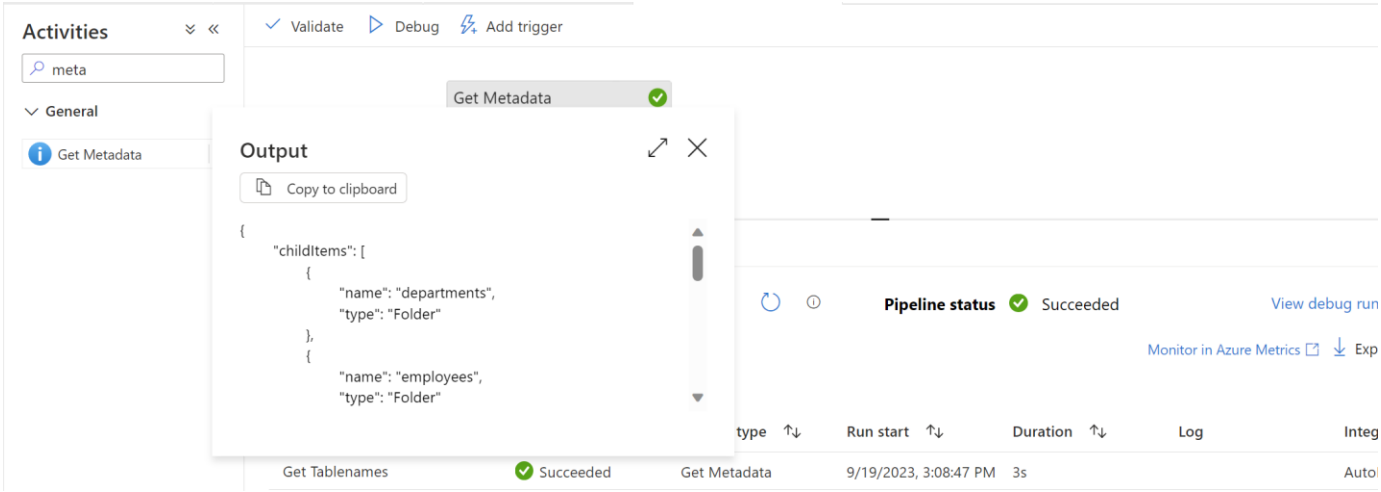
Create a New Pipeline – Idea is to create the tables first (using Metadata activity), then for each (ForEach activity) table load it with the dynamic query.



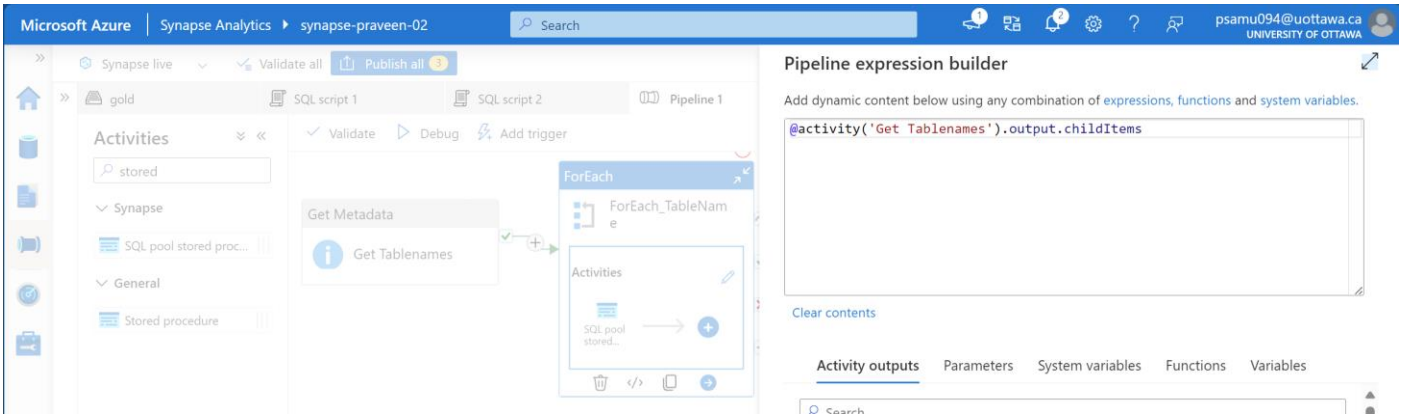
Create a dataset for the activity -> ADLS -> Binary format, choose the default Synapse connection to ADLS.



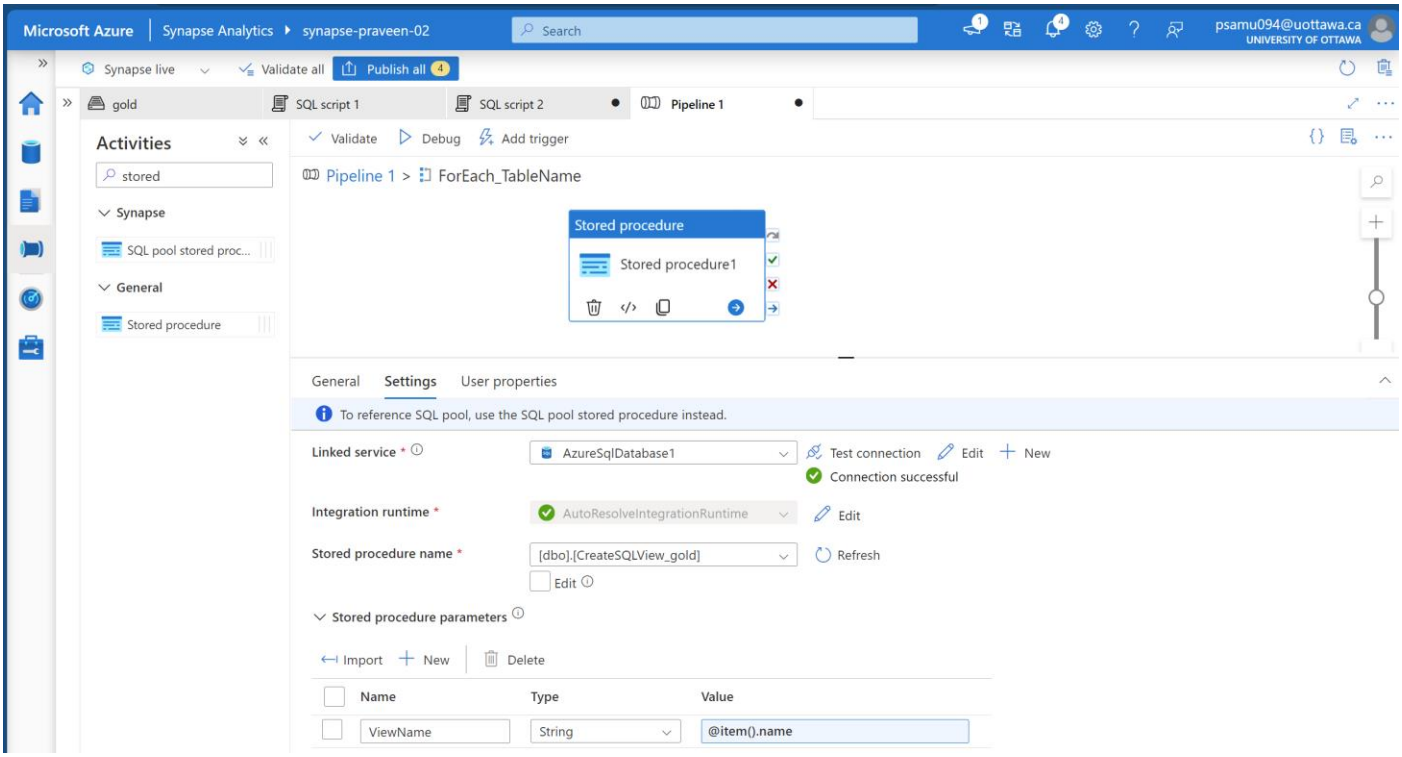
The result from Metadata will be as follows returning JSON data of items present in the location



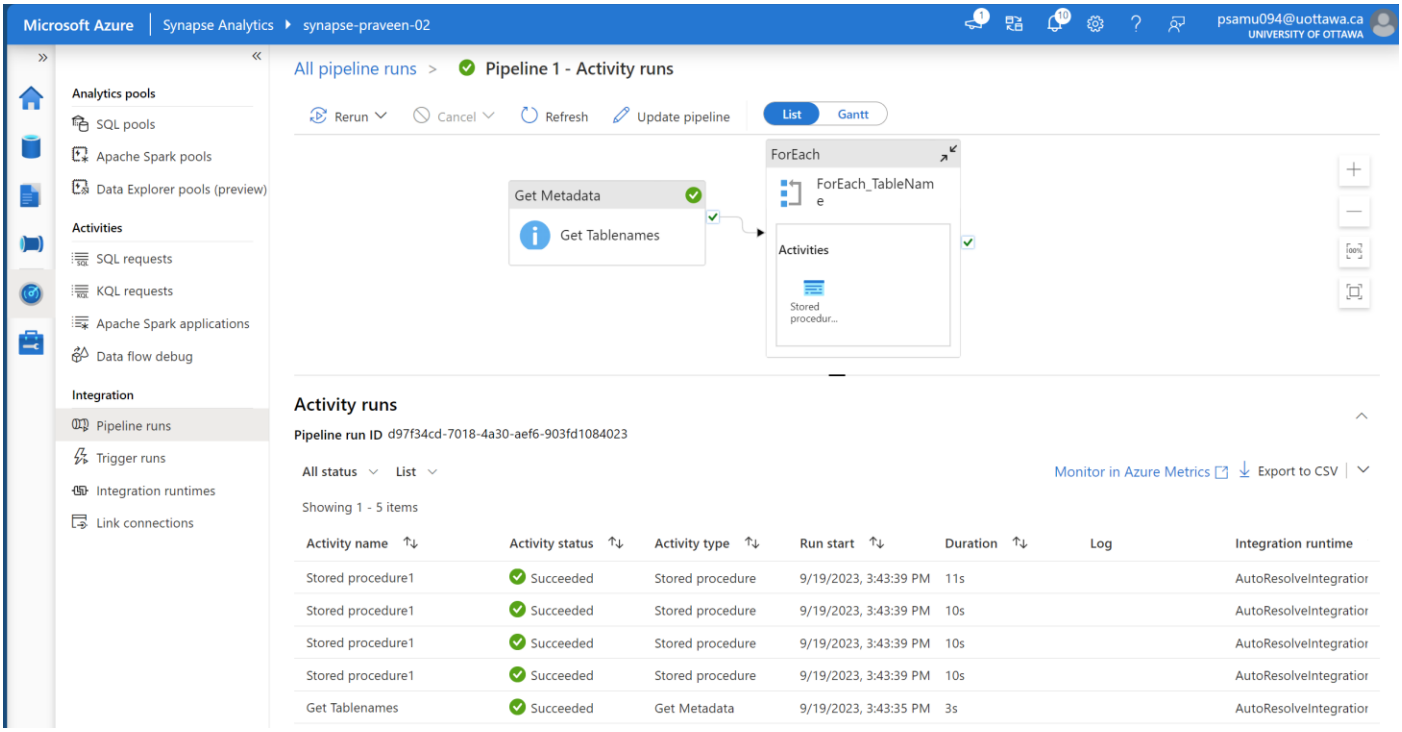
Insert ForEach activity as next step in pipeline and write dynamic SQL to fetch childItems from metadata result



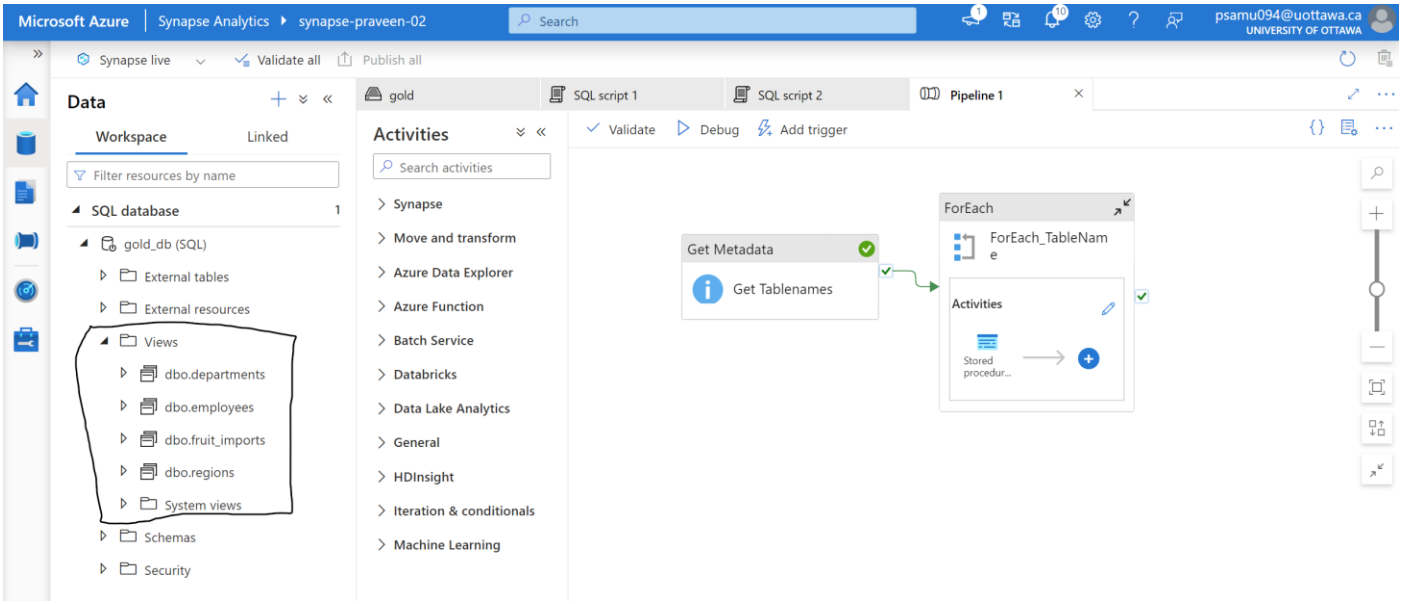
In the ForEach activity, place a Stored Procedure activity and configure the parameters



Run the pipeline and check if everything works as intended



Views are created



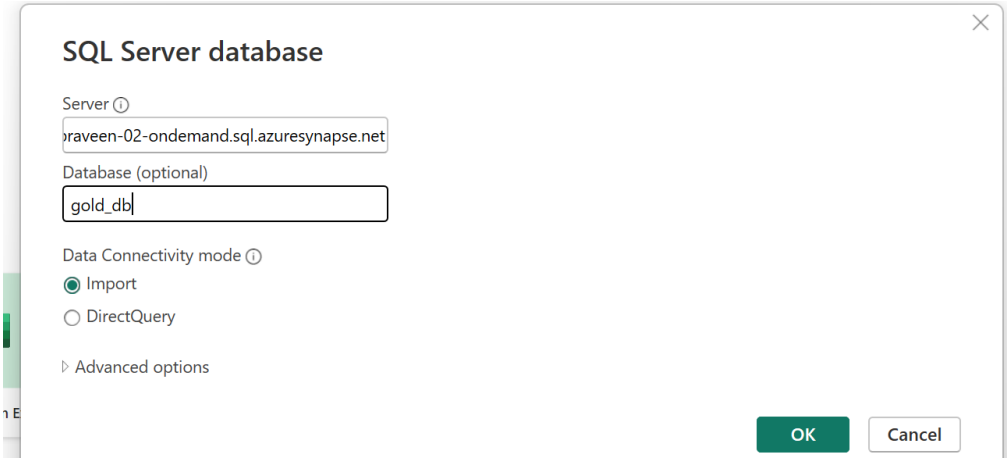
This pipeline is needed to run only when there is change in table schema at source table. Any newly added data will sync aytumatically.

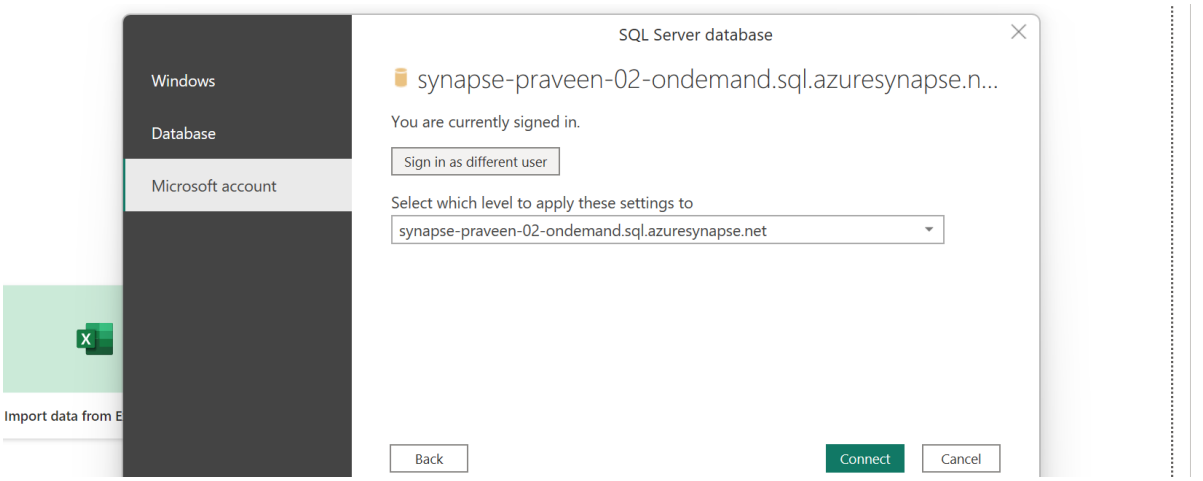
Step 6 – Provide Business Analytics using PowerBI with data in Synapse DB

Launch PowerBI desktop application

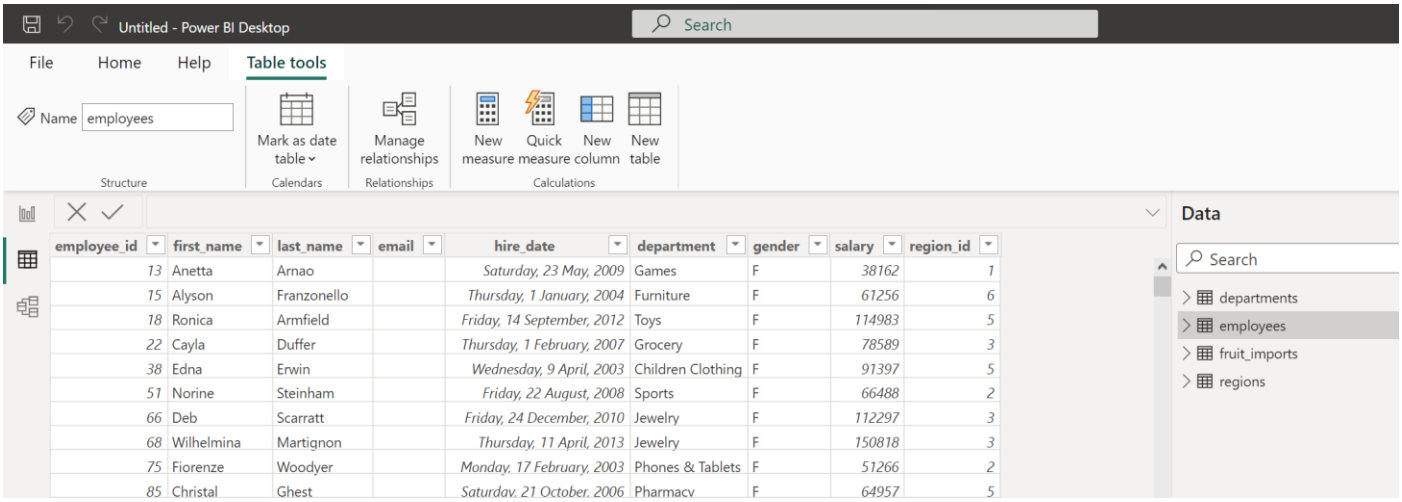
Get Data -> More -> Azure -> Synapse Analytics SQL -> Connect

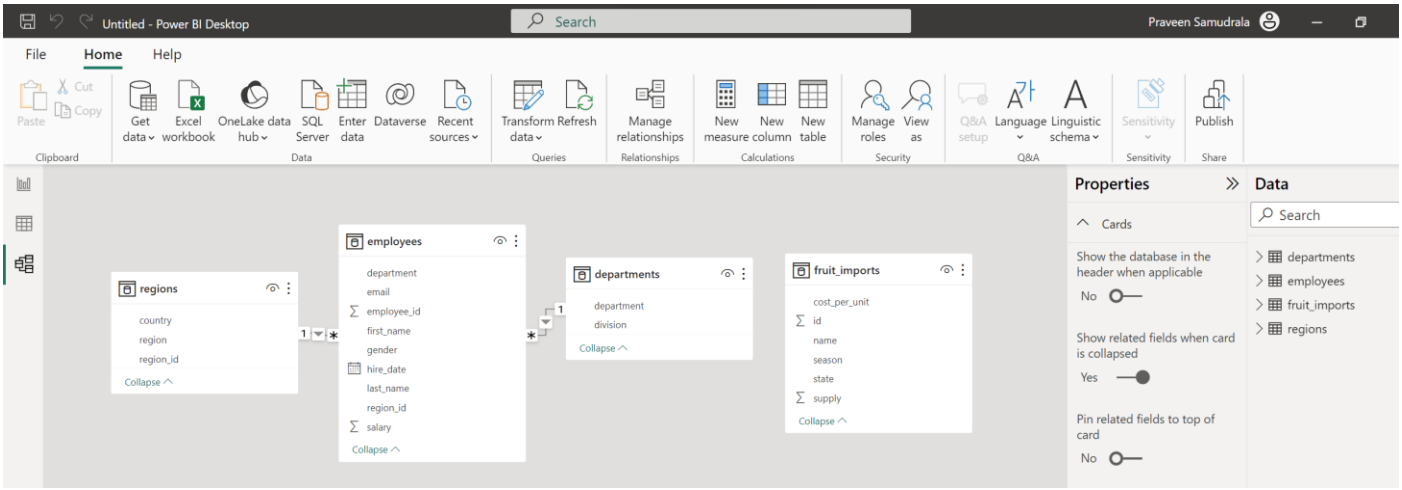
Get the Serverless SQL endpoint connection string from Synapse Properties and enter the details in PowerBI to connect to Synapse



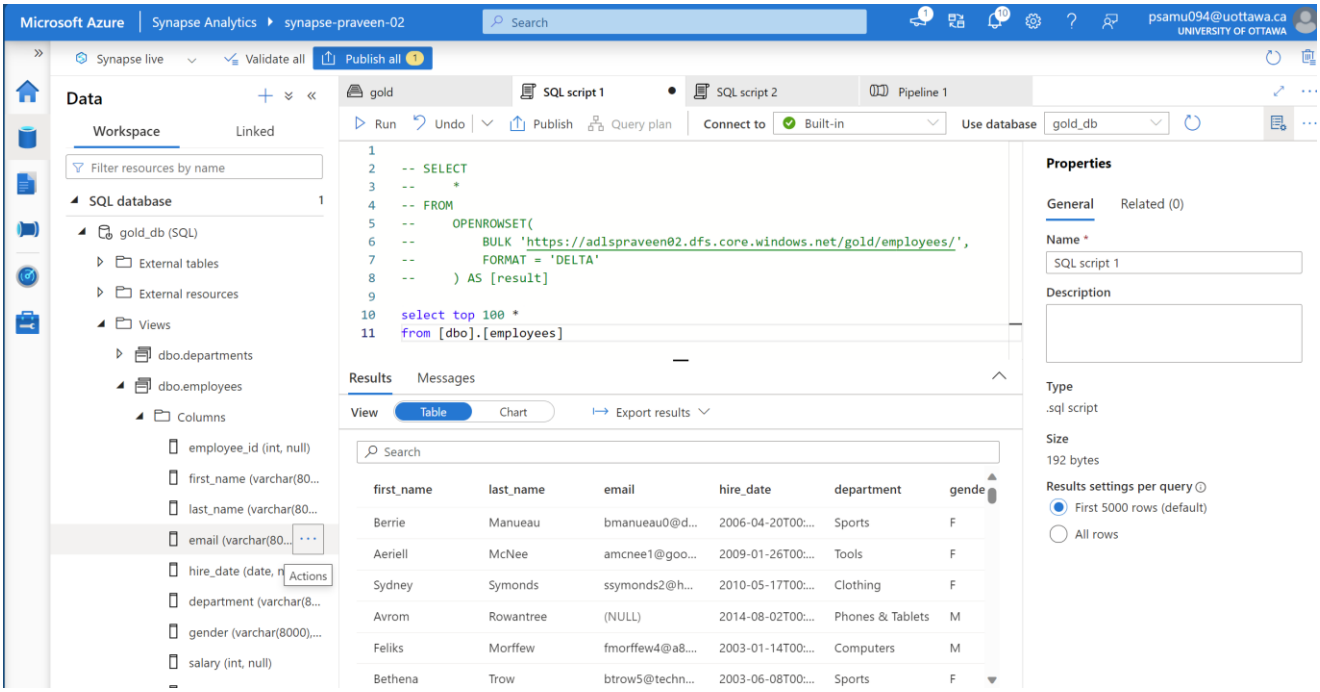


Choose the tables to be imported and choose ‘Load’.

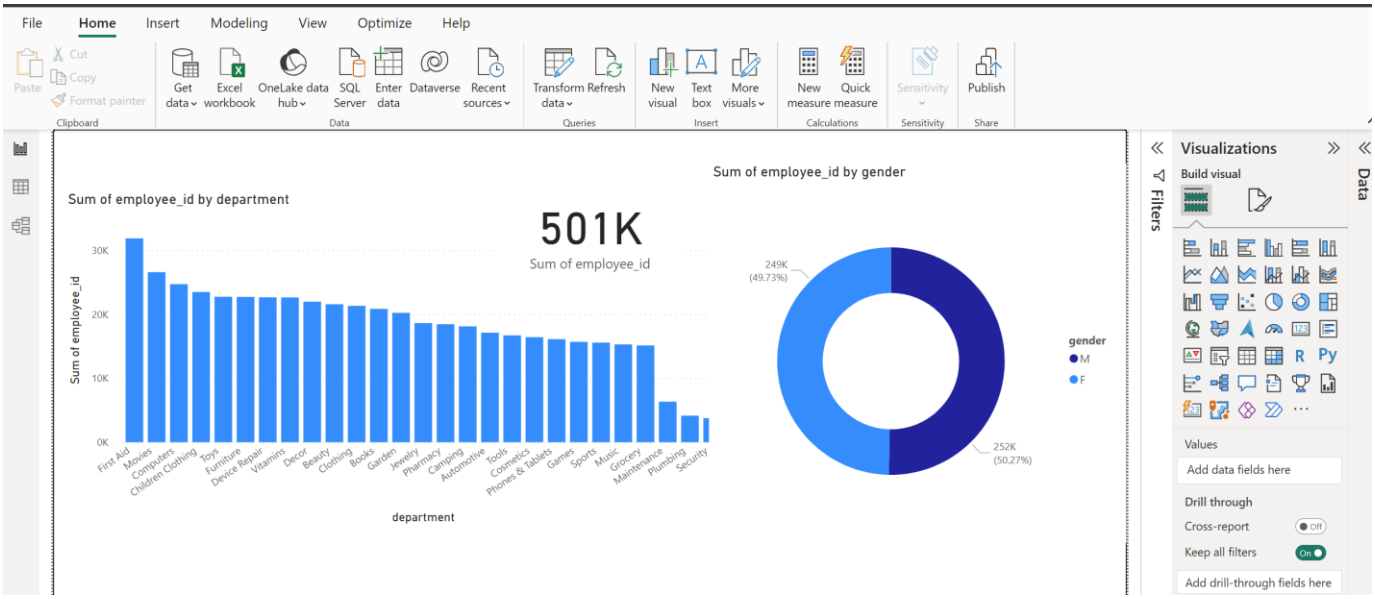


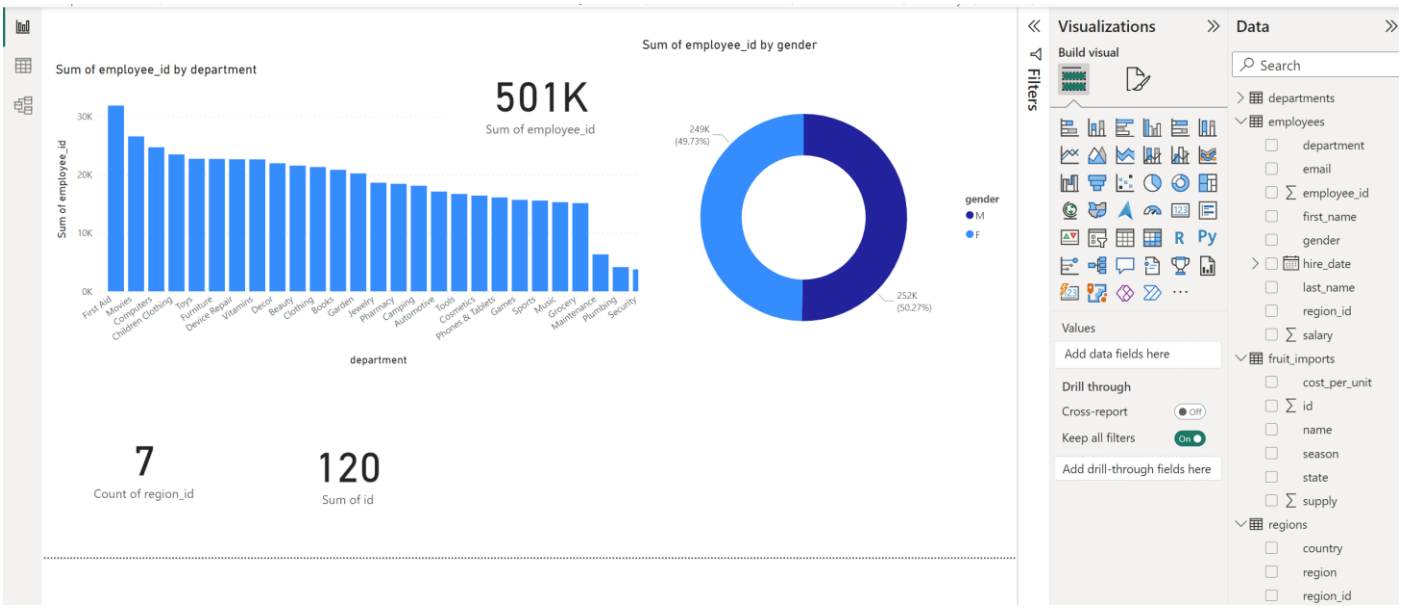


The values of email in employees table are available in Synapse DB views but not populating in PowerBI



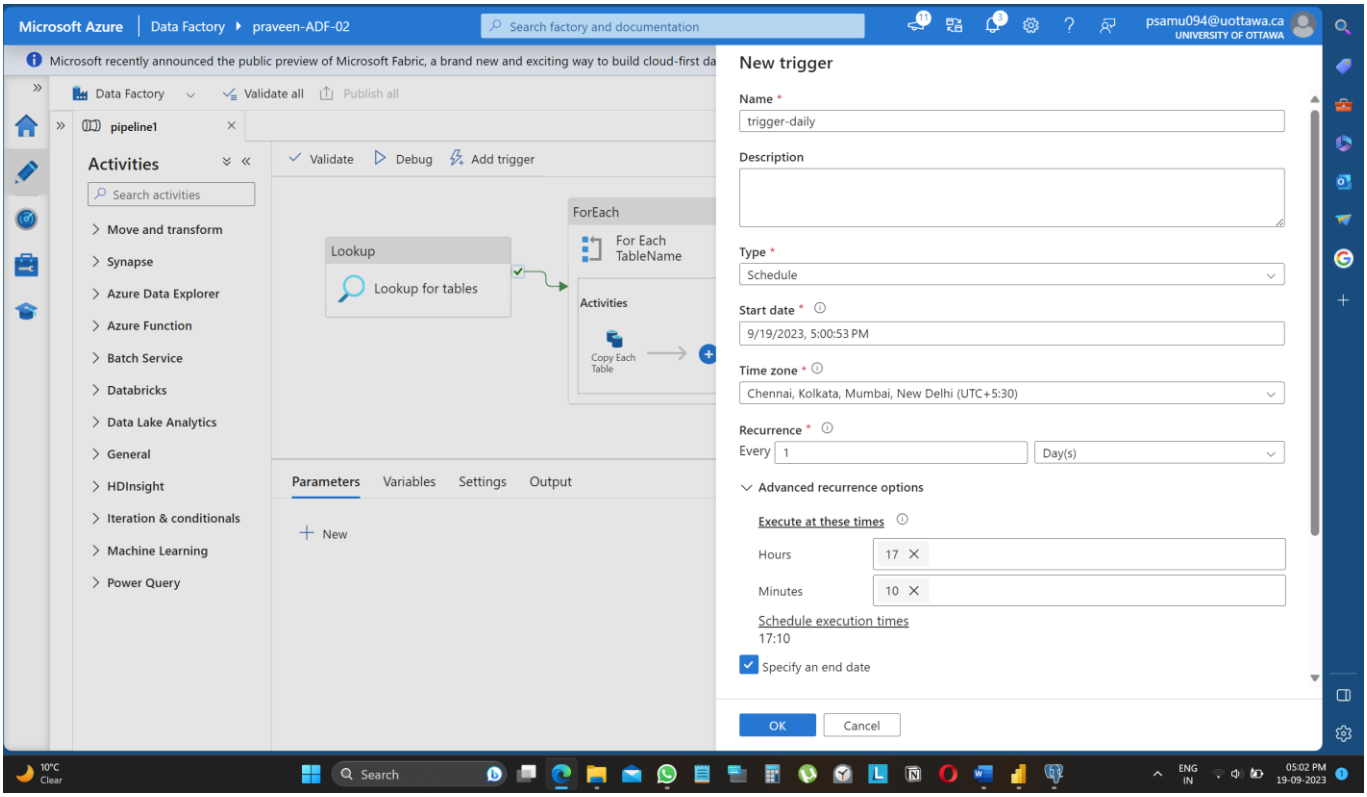
Visualizations in PowerBI


File
Home
Insert
Modeling
View
Optimize
Help
Sum of employee_id by department
501K
Sum of employee_id
Sum of employee_id by gender
department
gender
Visualizations
Build visual
Values
Add data fields here
Drill through
Cross-report
Keep all filters
Add drill-through fields here
Filters
Data

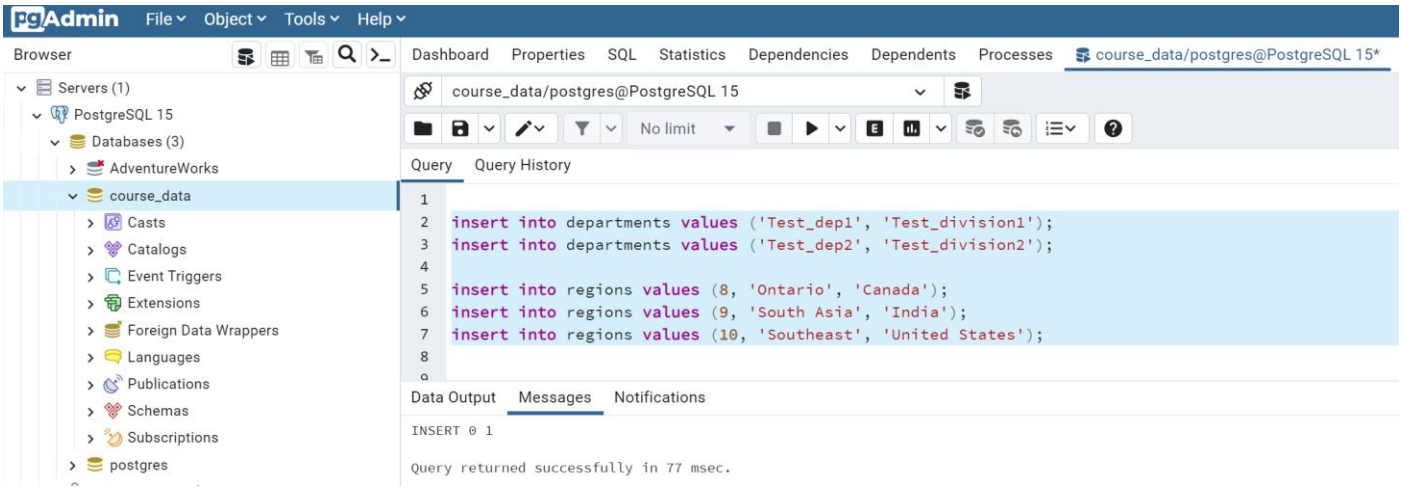

Sum of employee_id by department
501K
Sum of employee_id
Sum of employee_id by gender
department
gender
7
Count of region_id
120
Sum of id
Visualizations
Build visual
Values
Add data fields here
Drill through
Cross-report
Keep all filters
Add drill-through fields here
Filters
Data
Search
departments
employees
department
email
employee_id
first_name
gender
hire_date
last_name
region_id
salary
fruit_imports
cost_per_unit
id
name
season
state
supply
regions
country
region
region_id

Step 7 – Scheduling the Data Pipeline to run once a day

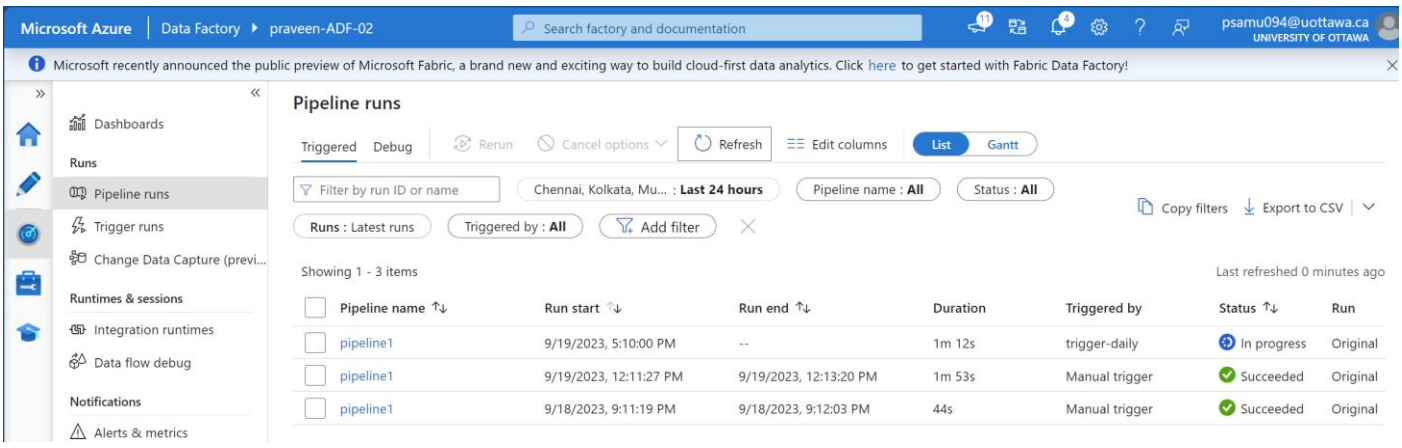
Schedule trigger for Azure Data Factory Pipeline



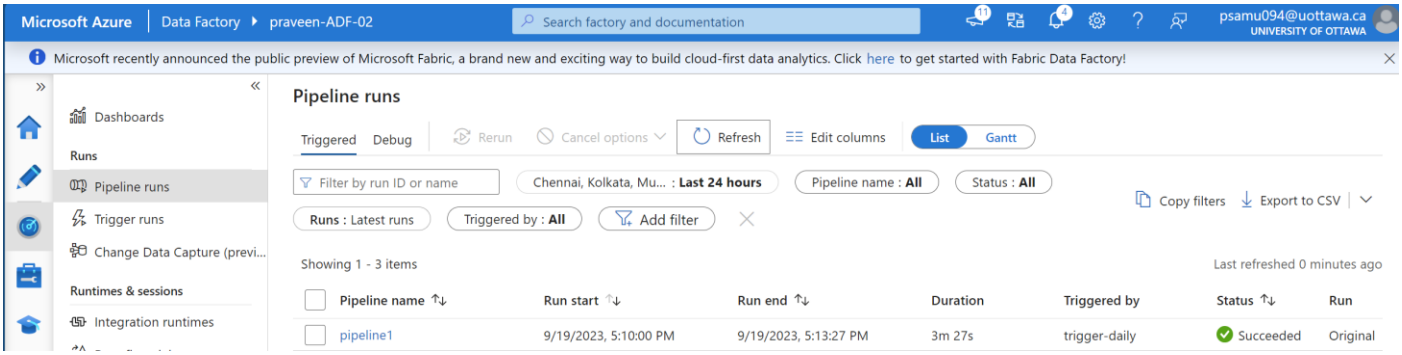
Added new values into tables



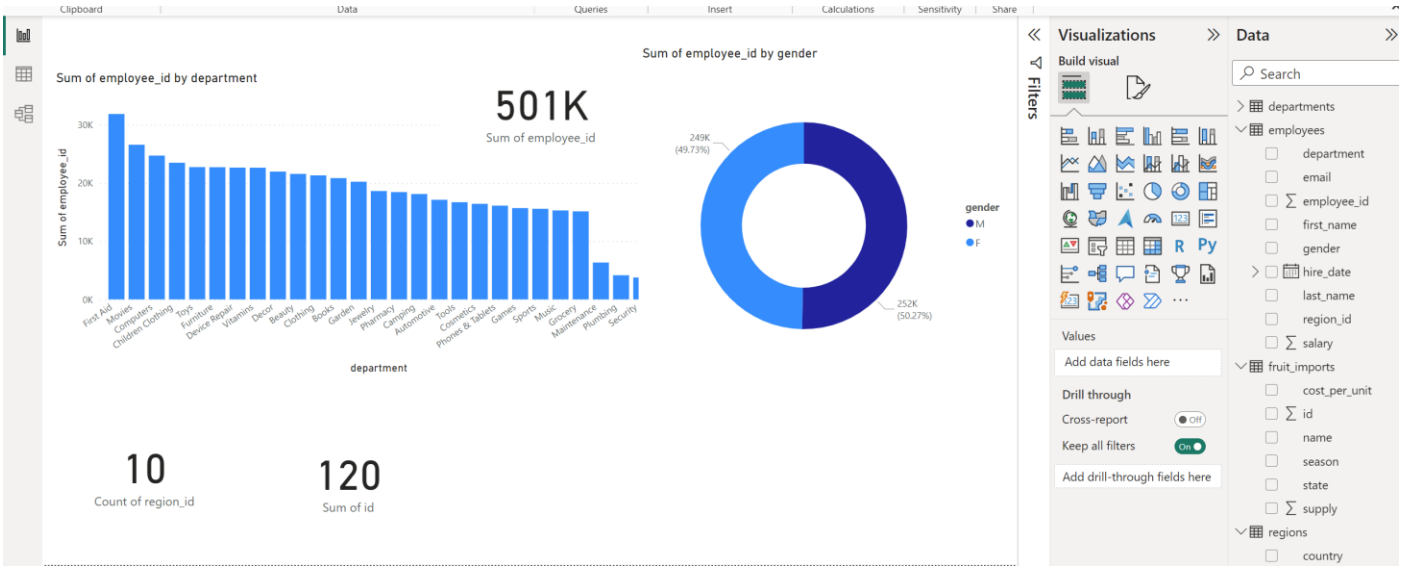
Pipeline execution at scheduled time



Pipeline completed successfully



Reflection of new data in PowerBI



Regions count updated from 7 to 10 after adding 3 new regions in on-premise PostgreSQL server.

References:

[An End to End Azure Data Engineering Real Time Project Demo | Get Hired as an Azure Data Engineer - YouTube]