

Azure Data Engineering Project – Adventure Works Sample Data

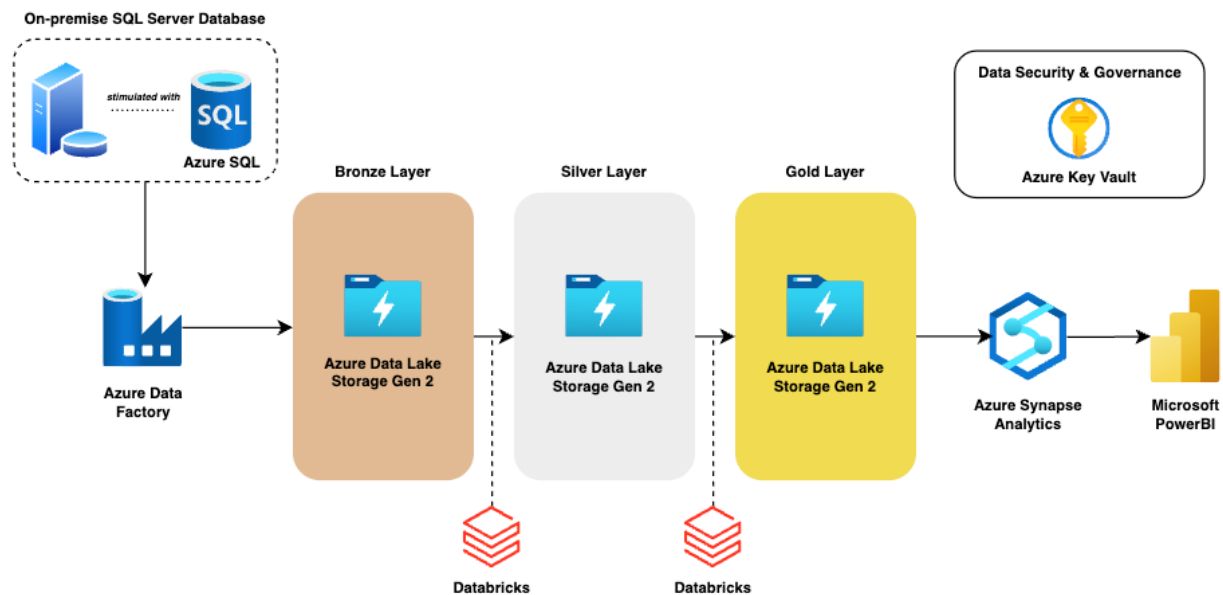
Overview:

This project aims to create an end-to-end data pipeline from ingestion data to visualization by leveraging various Azure services with the Adventure Works sample data. The pipeline will be triggered in a daily basis, and the latest updated data will be reflected in the dashboard automatically.

Data Source:

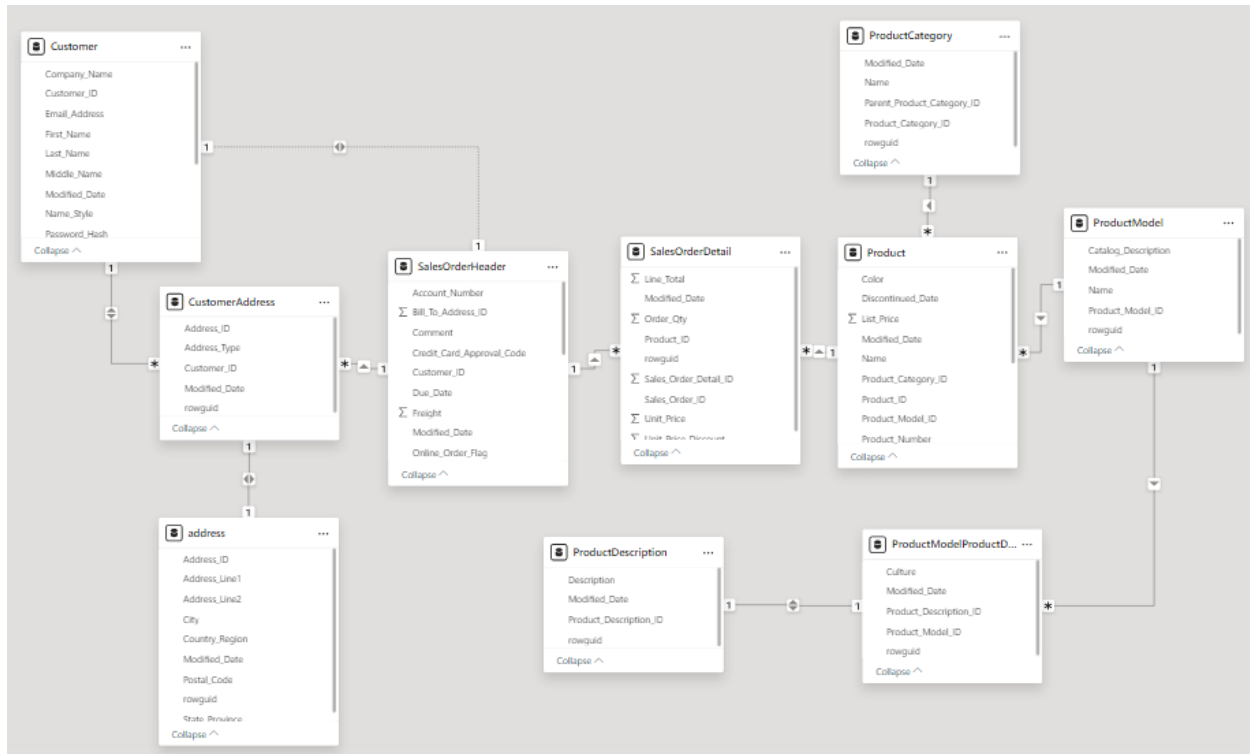
[Adventure Works Sample Data](#)

Data Pipeline Architecture:



- On-premises Database Simulation: Azure SQL, Microsoft SQL Server, Docker, Azure Data Studio
- Data Ingestion: Azure Data Factory
- Data Transformation: Azure Databricks
- Data Loading: Azure Synapse Analytics
- Data Visualization: Microsoft PowerBI
- Data Storage: Azure SQL, Azure Data Lake Storage Gen 2
- Data Security & Governance: Azure Key Vault

Data Model / Schema:



Setup: Restoring the database in MS SQL

*It would not work for Mac user as it cannot be connected to azure data factory due to the limitation of azure data studio (self-hosted integration runtime N/A)

1. Use docker to run MS SQL and Azure Data Studio as the replacement of SSMS due to the nature of Mac environment.
2. Run the below command to stimulate the on-premises database environment.
 - docker run -e "ACCEPT_EULA=1" -e "MSSQL_SA_PASSWORD={YOURPASSWORD}" -e "MSSQL_USER=SA" -p 1433:1433 -d --name=mssql mcr.microsoft.com/azure-sql-edge
3. Restore database by using the bak file provided in the data source website.
4. Create a login user and password for azure access to the database by using the command below.
 - CREATE USER <Azure_AD_principal_name> FROM EXTERNAL PROVIDER;
 - ALTER ROLE db_owner ADD MEMBER <ADF_service_name>
5. Store the credentials in Azure Key Vault for future usage.

Setup: Restore the database in Azure SQL database

1. Azure SQL database is used to stimulate the on-premises database and connect to azure data studio due to the limitation of Mac environment.
2. Select Azure SQL as the service and include the sample data in the configuration settings.
3. Write a T-SQL query to grant access to Azure Data Factory with the command below.
 - CREATE USER [de-adworks-adf] FROM EXTERNAL PROVIDER;
 - ALTER ROLE db_owner ADD MEMBER [de-adworks-adf];
4. Query the data for testing

The screenshot shows the Azure Data Studio interface. On the left, the 'Tables' folder is expanded, showing a list of tables including 'SalesLT.Address', 'SalesLT.Customer', 'SalesLT.CustomerAddress', 'SalesLT.Product', 'SalesLT.ProductCategory', 'SalesLT.ProductDescription', 'SalesLT.ProductModel', 'SalesLT.ProductModelProductDescription', 'SalesLT.SalesOrderDetail', 'SalesLT.SalesOrderHeader', and 'Dropped Ledger Tables'. The main editor displays a T-SQL query:

```
1 SELECT s.name AS SchemaName,
2 t.name as TableName
3 FROM sys.tables t
4 INNER JOIN sys.schemas s
5 ON t.schema_id = s.schema_id
6 WHERE s.name = 'SalesLT'
```

Below the query, the 'Results' tab is active, showing a table with two columns: 'SchemaName' and 'TableName'. The results are as follows:

	SchemaName	TableName
1	SalesLT	Customer
2	SalesLT	ProductModel
3	SalesLT	ProductDescription
4	SalesLT	Product
5	SalesLT	ProductModelProductDescription
6	SalesLT	ProductCategory
7	SalesLT	Address

At the bottom of the interface, the 'TASKS' tab is selected, showing 'No task history to display.'

Data Ingestion: Ingest Data from Azure SQL (act as on-prem database) to Bronze Layer Azure Data Lake Storage Gen 2

1. Add Azure Data Factory into the resource group
2. Connect Azure SQL database with self-hosted integration runtime in ADF
3. Creating a copy activity for testing purpose with only Address table.
4. Create a linked service to Azure SQL database with the proper authentication
5. Preview Address Data in Copy activity

Activities

Search activities

Move and transform

Copy data

Copy Address Table

Preview data

Linked service: azure_sqlserver

Object: SalesLT.Address

	AddressID	AddressLine1	AddressLine2	City	StateProvince	CountryRegion	PostalCode	rowguid
1	9	8713 Yosemite Ct.		Bothell	Washington	United States	98011	268af76d7-9441-144fd
2	11	1318 Lasalle Street		Bothell	Washington	United States	98011	981b:aca2-9a96-fb670
3	25	9178 Jumping St.		Dallas	Texas	United States	75201	c8df348f0-a8dd-14a67
4	28	9228 Via Del Sol		Phoenix	Arizona	United States	85004	12ae5fc3e-49b92-3b970
		76010 Indefa						84a957a0R..

Please preview data to validate the partition settings.

Additional columns

New

Preview data

6. Configure sink to ADLS by creating a storage account for ADLS and create a bronze container to store the raw data file
7. Test the copy activity to check if address table is successfully ingested into bronze table in ADLS.
8. Create a Copy All Tables Pipeline that ingest all tables in Azure SQL database into the bronze layer.
9. Add Lookup activity in pipeline to query the available tables in the database by using SQL query
 - SELECT s.name AS SchemaName,
t.name as TableName
FROM sys.tables t
INNER JOIN sys.schemas s
ON t.schema_id = s.schema_id
WHERE s.name = 'SalesLT'
 - The output of the lookup activity is used as a parameter to perform the copy of all tables

General
Settings
User properties

Source dataset *
AzureSQLDB
Open
New
Pr

First row only
☐

Use query
☐ Table
☒ Query
☐ Stored procedure

Query *
SELECT s.name AS SchemaName,
t.name as TableName
FROM sys.tables t
Edit

Query timeout (minutes) ⓘ
120

Isolation level ⓘ
Select...

Partition option ⓘ
☒ None
☐ Physical partitions of table ⓘ
☐ Dynamic range ⓘ

Please preview data to validate the partition settings.

10. Add for each activity with by wrapping a copy data activity inside

copy_pipeline
Validate
Debug
Add trigger

Lookup
Look for all tables

General
Settings
Activities (1)
Us

Sequential
☐

Batch count ⓘ

Items
@activ

Pipeline expression builder
Add dynamic content below using any combination of expressions, functions and system variables.
@activity('Look for all tables').output.value
Clear contents

Activity outputs
Parameters
System variables
Functions
Variables

Search

Copy Each Table
Copy Each Table activity output

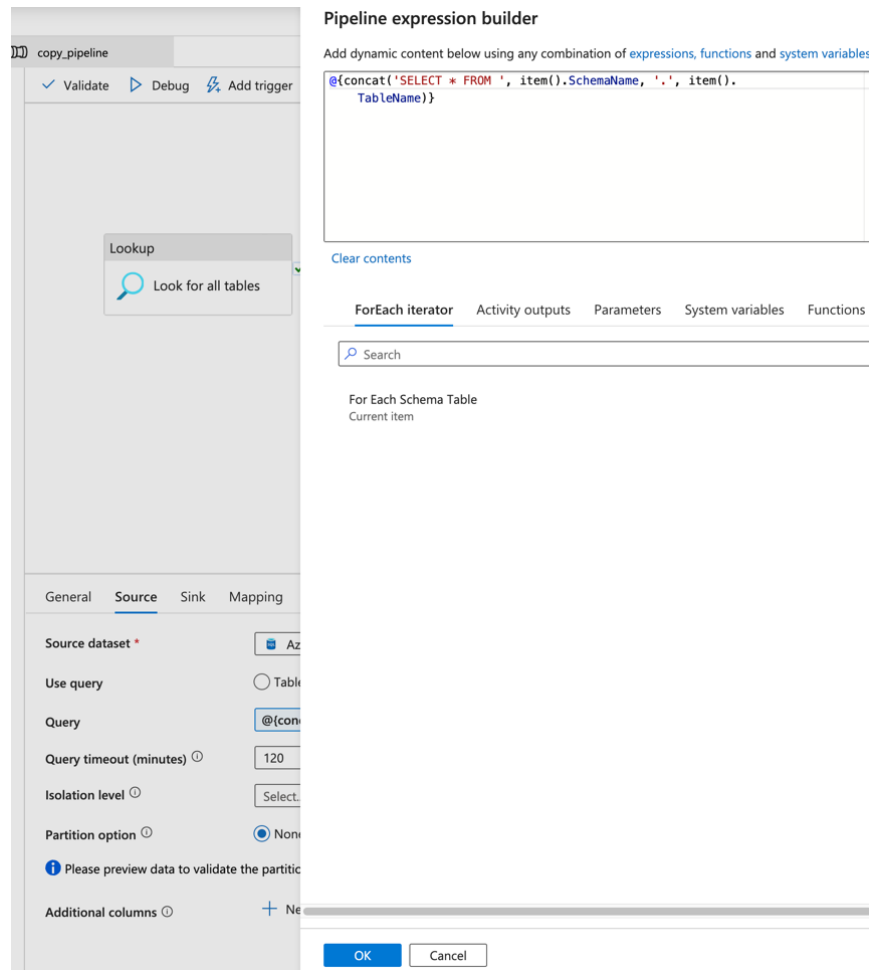
Copy Each Table
Copy Each Table pipeline return value

Look for all tables
Look for all tables activity output

Look for all tables
Look for all tables pipeline return value

Look for all tables count
Count of the rows

Look for all tables value array
Array of row data



11. Configure folder structure for bronze storage as the sink

- bronze/schemaName/tableName/table.parquet

12. Test the pipeline with trigger run

All pipeline runs > ✔ copy_all_tables - Activity runs

[Rerun](#) [Cancel](#) [Refresh](#) [Update pipeline](#) [List](#) [Gantt](#)

i Pipeline was modified after this run. The current pipeline configuration is shown.

Activity runs

Pipeline run ID 3be99536-2299-47b1-98c0-91e260c413e5

All status ▼ [Monitor in Azure Metrics](#) [Export to CSV](#)

Showing 1 - 12 items

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	User properties	Activity run ID
Copy Each Table	✔ Succeeded	Copy data	6/20/2024, 7:28:39 PM	19s	AutoResolveIntegrator		91d560b8-0255-
Copy Each Table	✔ Succeeded	Copy data	6/20/2024, 7:28:39 PM	19s	AutoResolveIntegrator		f7c2cccc-5b57-4-
Copy Each Table	✔ Succeeded	Copy data	6/20/2024, 7:28:39 PM	20s	AutoResolveIntegrator		a3e8715e-b647--
Copy Each Table	✔ Succeeded	Copy data	6/20/2024, 7:28:39 PM	19s	AutoResolveIntegrator		c8f38557-d13e-4
Copy Each Table	✔ Succeeded	Copy data	6/20/2024, 7:28:39 PM	19s	AutoResolveIntegrator		8c6fd128-d458-4
Copy Each Table	✔ Succeeded	Copy data	6/20/2024, 7:28:39 PM	19s	AutoResolveIntegrator		6c78e006-bf0b-4
Copy Each Table	✔ Succeeded	Copy data	6/20/2024, 7:28:39 PM	18s	AutoResolveIntegrator		91a102f2-aa51-4
Copy Each Table	✔ Succeeded	Copy data	6/20/2024, 7:28:39 PM	18s	AutoResolveIntegrator		5a3d9868-b8b0-
Copy Each Table	✔ Succeeded	Copy data	6/20/2024, 7:28:39 PM	20s	AutoResolveIntegrator		e93ff37a-02df-4;
Copy Each Table	✔ Succeeded	Copy data	6/20/2024, 7:28:39 PM	18s	AutoResolveIntegrator		ca052c86-54ac-4
For Each Schema Table	✔ Succeeded	ForEach	6/20/2024, 7:28:38 PM	23s			dd8c4735-ef8e-4
Look for all tables	✔ Succeeded	Lookup	6/20/2024, 7:28:31 PM	7s	adworks-shir (Southeas		b2a9ff5a-1b68-4

Data Transformation:

1. Add Azure Databricks into the resource group
2. Mount ADLS to databricks to get access to the data
 - not necessary to mount the data since the azure credential passthrough is enabled in the cluster, but it will ease the coding part as different containers have different mount points.
3. Create a notebook to mount the data

```
▶ ✓ 2 days ago (11s) 4

1 configs = {
2     "fs.azure.account.auth.type": "CustomAccessToken",
3     "fs.azure.account.custom.token.provider.class": spark.conf.get("spark.databricks.passthrough.adls.gen2.
4     tokenProviderClassName")
5 }
6 # Optionally, you can add <directory-name> to the source URI of your mount point.
7 dbutils.fs.mount(
8     source = "abfss://silver@adworksadls.dfs.core.windows.net/",
9     mount_point = "/mnt/silver",
10    extra_configs = configs)

True
```

```
⋮ ▶ ✓ 2 days ago (11s) 5 Python ✨ ⌂ ⋮ 🗑️
▼
1 configs = {
2     "fs.azure.account.auth.type": "CustomAccessToken",
3     "fs.azure.account.custom.token.provider.class": spark.conf.get("spark.databricks.passthrough.adls.gen2.
4     tokenProviderClassName")
5 }
6 # Optionally, you can add <directory-name> to the source URI of your mount point.
7 dbutils.fs.mount(
8     source = "abfss://gold@adworksadls.dfs.core.windows.net/",
9     mount_point = "/mnt/gold",
10    extra_configs = configs)

True
```

4. [Bronze → Silver] Transform date column into date format as the date columns in every table is in datetime format in default
5. Run a loop to transform all the files in bronze layer and save it in silver layer in delta format

bronze to silver Python ☆
File Edit View Run Help Last edit was yesterday Provide feedback

Run all Terminated Schedule

Transformation for all tables

Yesterday (<1s) 8

```
1 table_name = []
2
3 for i in dbutils.fs.ls('/mnt/bronze/SalesLT'):
4     table_name.append(i.name.split('/')[0])
```

Yesterday (<1s) 9

```
1 table_name
[Address',
Customer',
CustomerAddress',
Product',
ProductCategory',
ProductDescription',
ProductModel',
ProductModelProductDescription',
SalesOrderDetail',
SalesOrderHeader']
```

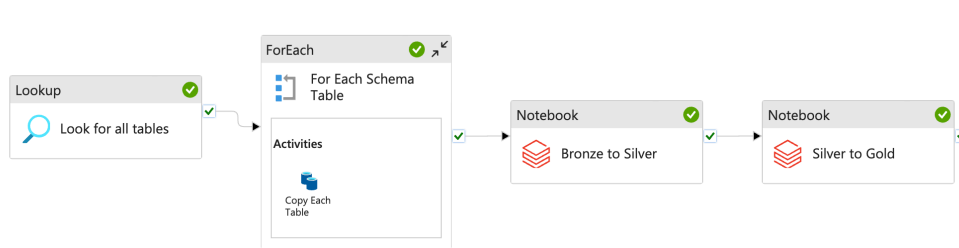
Yesterday (55s) 10

```
1 from pyspark.sql.functions import from_utc_timestamp, date_format
2 from pyspark.sql.types import TimestampType
3
4 for i in table_name:
5     path = '/mnt/bronze/SalesLT/' + i + '/' + i + '.parquet'
6     df = spark.read.format('parquet').load(path)
7     column = df.columns
8
9     for col in column:
10         if "Date" in col or "date" in col:
11             df = df.withColumn(col, date_format(from_utc_timestamp(df[col].cast(TimestampType()), "UTC"), "yyyy-MM-dd"))
12
13     output_path = '/mnt/silver/SalesLT/' + i + '/'
14     df.write.format('delta').mode('overwrite').save(output_path)
```

▶ (64) Spark Jobs

6. [Silver → Gold] Convert every column naming format from ColumnName to Column_Name for better naming conventions

9. Test pipeline and verify if the files are successfully transformed and loaded into ADLS.



Activity runs

Pipeline run ID e8175844-456a-495c-8c02-ce2aeb4c8f60

All status List

[Monitor in Azure Metrics](#) [Export to CSV](#)

Showing 1 - 14 items

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	User properties	Activity run ID
Copy Each Table	Succeeded	Copy data	6/21/2024, 10:22:39 PM	10s	AutoResolveIntegration		31106530-7111
Copy Each Table	Succeeded	Copy data	6/21/2024, 10:22:39 PM	20s	AutoResolveIntegration		8a60574e-336
Copy Each Table	Succeeded	Copy data	6/21/2024, 10:22:39 PM	18s	AutoResolveIntegration		91027e70-eae
Copy Each Table	Succeeded	Copy data	6/21/2024, 10:22:39 PM	18s	AutoResolveIntegration		d89c87bc-ea3
Copy Each Table	Succeeded	Copy data	6/21/2024, 10:22:39 PM	20s	AutoResolveIntegration		9712870a-3d5
Copy Each Table	Succeeded	Copy data	6/21/2024, 10:22:39 PM	19s	AutoResolveIntegration		71ab07c2-cd3
Bronze to Silver	Succeeded	Notebook	6/21/2024, 10:23:04 PM	6m 36s	AutoResolveIntegration		63381695-83e
Silver to Gold	Succeeded	Notebook	6/21/2024, 10:29:40 PM	1m 10s	AutoResolveIntegration		815547ad-ea3

silver

Container

Search

Upload Add Directory Refresh

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

Shared access tokens

Manage ACL

Access policy

Properties

Metadata

Authentication method: Access key (Switch to Microsoft Entra ID)
Location: silver / SalesLT

Search blobs by prefix (case-sensitive)

Name

- ☐ [-]
- ☐ Address
- ☐ Customer
- ☐ CustomerAddress
- ☐ Product
- ☐ ProductCategory
- ☐ ProductDescription
- ☐ ProductModel
- ☐ ProductModelProductDescription
- ☐ SalesOrderDetail
- ☐ SalesOrderHeader

Home > Resource groups > de-adworks-rg > adworksadls | Containers >

gold ...
Container

Search « Upload + Add Directory Refresh Rename

Overview
Diagnose and solve problems
Access Control (IAM)

Settings
Shared access tokens
Manage ACL
Access policy
Properties
Metadata

Authentication method: Access key ([Switch to Microsoft Entra user](#))
Location: gold / SalesLT

Search blobs by prefix (case-sensitive)

Name	Modified
<input type="checkbox"/> [..]	
<input type="checkbox"/> Address	
<input type="checkbox"/> Customer	
<input type="checkbox"/> CustomerAddress	
<input type="checkbox"/> Product	
<input type="checkbox"/> ProductCategory	
<input type="checkbox"/> ProductDescription	
<input type="checkbox"/> ProductModel	
<input type="checkbox"/> ProductModelProductDescription	
<input type="checkbox"/> SalesOrderDetail	
<input type="checkbox"/> SalesOrderHeader	

Data Loading: Load the gold table into Azure Synapse to create views for each table

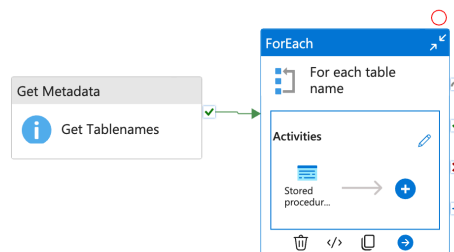
1. Add Azure Synapse into the resource group
 - Azure Synapse built on top of ADF and have more features than it such as creating SQL or Spark notebooks)
2. Create a serverless SQL pool and an Azure SQL database to store all the views
 - Serverless SQL pool come in default, it does not have cost for storage, direct query from ADLS and suitable for smaller workload
 - Dedicated SQL pool need to create a cluster like databricks, it has cost for storage and compute, will load the data into the compute, but have better performance
3. Create a stored procedure to create a view for every tables

```

1  USE gold_db
2  GO
3
4  CREATE OR ALTER PROC CreateSQLServerlessView_gold @ViewName NVARCHAR(100)
5  AS
6  BEGIN
7
8  DECLARE @statement VARCHAR(MAX)
9
10     SET @statement = N'CREATE OR ALTER VIEW ' + @ViewName + ' AS
11         SELECT *
12         FROM
13             OPENROWSET(
14                 BULK 'https://adworksadls.dfs.core.windows.net/gold/SalesLT/' + @ViewName + '/',
15                 FORMAT = 'DELTA'
16             ) as [result]
17
18     EXEC (@statement)
19
20 END
21
22
23 Gd

```

4. Establish linked service to the gold_db for the stored procedure to access to the gold database
5. Create a pipeline to run the stored procedure
6. Add Get Metadata activity to get all the tables in the gold layer
7. Add for each activity by wrapping a notebook activity inside to run the stored procedure for each table



General Settings User properties

To reference SQL pool, use the SQL pool stored procedure instead.

Linked service * Test connection Edit + New

Integration runtime * Edit

Stored procedure name * Enter manually

Stored procedure parameters

Import + New Delete

Name	Type	Value
<input type="checkbox"/> ViewName	String	@item().name

8. Run and test the pipeline
 - Only run the pipeline when there is a schema change

The screenshot shows an Azure Data Factory pipeline with two main activities: 'Get Metadata' (Get Tablenames) and a 'ForEach' loop. The 'ForEach' loop is configured to iterate 'For each table name' and contains an 'Activities' container with a 'Stored procedure' activity. Below the pipeline diagram, the 'Activity runs' section for Pipeline run ID 8803d7be-12ad-4a9b-84de-2dd766db541b is displayed. It shows a list of 12 items, all of which are 'Stored procedure1' activities that have 'Succeeded'.

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	User properties	Activity run ID
Stored procedure1	Succeeded	Stored procedure	6/21/2024, 11:22:33 PM	11s	AutoResolveIntegration		91ffa7c-e3
Stored procedure1	Succeeded	Stored procedure	6/21/2024, 11:22:33 PM	11s	AutoResolveIntegration		9bc94e7d-e
Stored procedure1	Succeeded	Stored procedure	6/21/2024, 11:22:33 PM	18s	AutoResolveIntegration		a5aa038e-6
Stored procedure1	Succeeded	Stored procedure	6/21/2024, 11:22:33 PM	11s	AutoResolveIntegration		435f72f9-e
Stored procedure1	Succeeded	Stored procedure	6/21/2024, 11:22:33 PM	12s	AutoResolveIntegration		234170b2-c
Stored procedure1	Succeeded	Stored procedure	6/21/2024, 11:22:33 PM	29s	AutoResolveIntegration		f5062f28-9
Stored procedure1	Succeeded	Stored procedure	6/21/2024, 11:22:33 PM	12s	AutoResolveIntegration		960b9dd1-
Stored procedure1	Succeeded	Stored procedure	6/21/2024, 11:22:33 PM	12s	AutoResolveIntegration		d70cd66-8

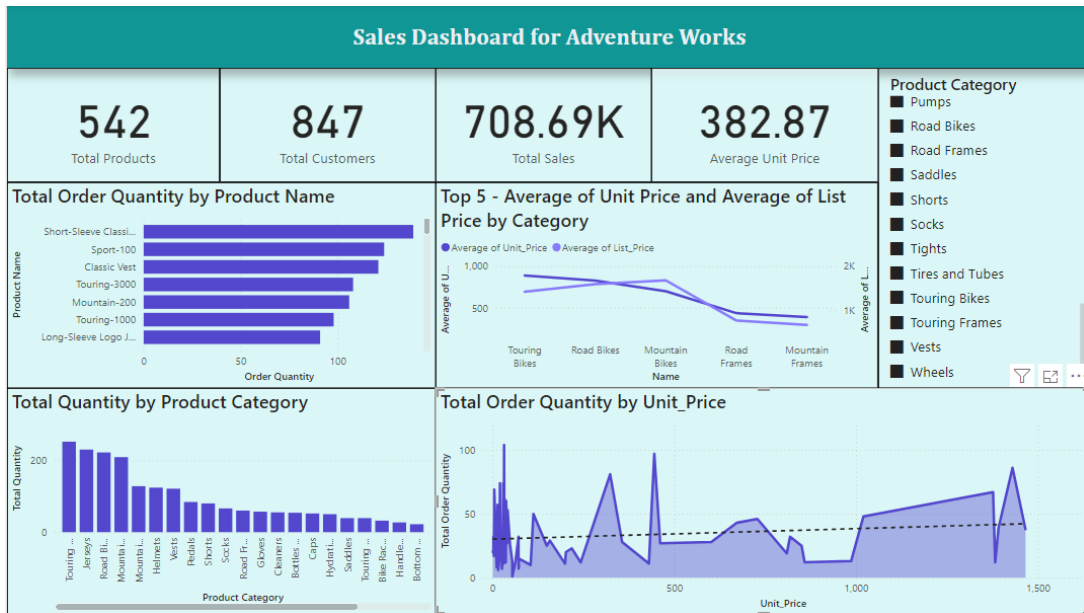
9. Verify if the views are created.

The screenshot shows the 'SQL database' explorer in SQL Server Enterprise Manager. The 'gold_db (SQL)' database is selected, and the 'Views' folder is expanded, showing a list of 12 views: dbo.address, dbo.Customer, dbo.CustomerAddress, dbo.Product, dbo.ProductCategory, dbo.ProductDescription, dbo.ProductModel, dbo.ProductModelProductD..., dbo.SalesOrderDetail, dbo.SalesOrderHeader, System views, Schemas, and Security.

Data Visualization or Reporting:

1. Connect PowerBI to Azure Synapse as our gold tables views are stored there
2. Use Direct Query mode to import the data into PowerBI
 - Import Mode → Loads data into Power BI
 - Direct Query → Direct querying data from Azure Synapse

3. Connect the relationship between customer and address as the relationship is missing between the tables.
4. Create the interactive dashboard.



Completing the data pipeline: Connecting all the activities together

1. Add schedule trigger (daily trigger)
2. Add new rows to the Customer table in Azure SQL raw database
 - SET IDENTITY_INSERT [Adventure-Works-d].[SalesLT].[Customer] ON;

```
INSERT INTO [Adventure-Works-d].[SalesLT].[Customer](
    [CustomerID]
    ,[NameStyle]
    ,[Title]
    ,[FirstName]
    ,[MiddleName]
    ,[LastName]
    ,[Suffix]
    ,[CompanyName]
    ,[SalesPerson]
    ,[EmailAddress]
    ,[Phone]
    ,[PasswordHash]
    ,[PasswordSalt]
    ,[rowguid]
    ,[ModifiedDate])
```

VALUES

(5555, 0, 'Mr.', 'Lionel', 'Messi', 'U', 'Jr.', 'dsa', 'adventure-works\Depp', 'depp@gmail.com', '2134-432-4345', 'Software', 'Developer', '750f3495-59c4-46a0-80e1-e37ec60e77d9', '2006-08-01 00:00:00.000'),

(5964, 0, 'Mr.', 'Chris', 'Ronaldo', 'J', 'Jr.', 'rtyr', 'adventure-works\Seb', 'seb@gmail.com', '3689-314-2387', 'Infra', 'Engineer', '750f3495-59c4-48a0-80e1-e37ec60e77d8', '2006-08-01 00:00:00.000')

SET IDENTITY_INSERT [Adventure-Works-d].[SalesLT].[Customer] OFF;

Users > adam > Documents > Projects > azure-de > AzureDE_Query

Run Cancel Disconnect Change Database: Adventure-Works-d Estimated Plan Enable Actual Plan Parse Enable SQLCMD To Notebook

```
7
8 SET IDENTITY_INSERT [Adventure-Works-d].[SalesLT].[Customer] ON;
9
10 INSERT INTO [Adventure-Works-d].[SalesLT].[Customer](
11     [CustomerID]
12     , [NameStyle]
13     , [Title]
14     , [FirstName]
15     , [MiddleName]
16     , [LastName]
17     , [Suffix]
18     , [CompanyName]
19     , [SalesPerson]
20     , [EmailAddress]
21     , [Phone]
22     , [PasswordHash]
23     , [PasswordSalt]
24     , [rowguid]
25     , [ModifiedDate])
26 VALUES
27 (5555, 0, 'Mr.', 'Lionel', 'Messi', 'U', 'Jr.', 'dsa', 'adventure-works\Depp', 'depp@gmail.com', '2134-432-4345', 'Software', 'Developer', '750f3495-59c4-46a0-80e1-e37ec60e77d9', '2006-08-01 00:00:00.000'),
28 (5964, 0, 'Mr.', 'Chris', 'Ronaldo', 'J', 'Jr.', 'rtyr', 'adventure-works\Seb', 'seb@gmail.com', '3689-314-2387', 'Infra', 'Engineer', '750f3495-59c4-48a0-80e1-e37ec60e77d8', '2006-08-01 00:00:00.000')
29
30 SET IDENTITY_INSERT [Adventure-Works-d].[SalesLT].[Customer] OFF;
```

Messages

1:46:37 AM Started executing query at Line 8
(2 rows affected)
Total execution time: 00:00:00.032

3. Test and run the whole pipeline

de-adworks-adf Search factory and documentation 22119291@imail.sunway.edu.my SUNWAY EDUCATION GROUP

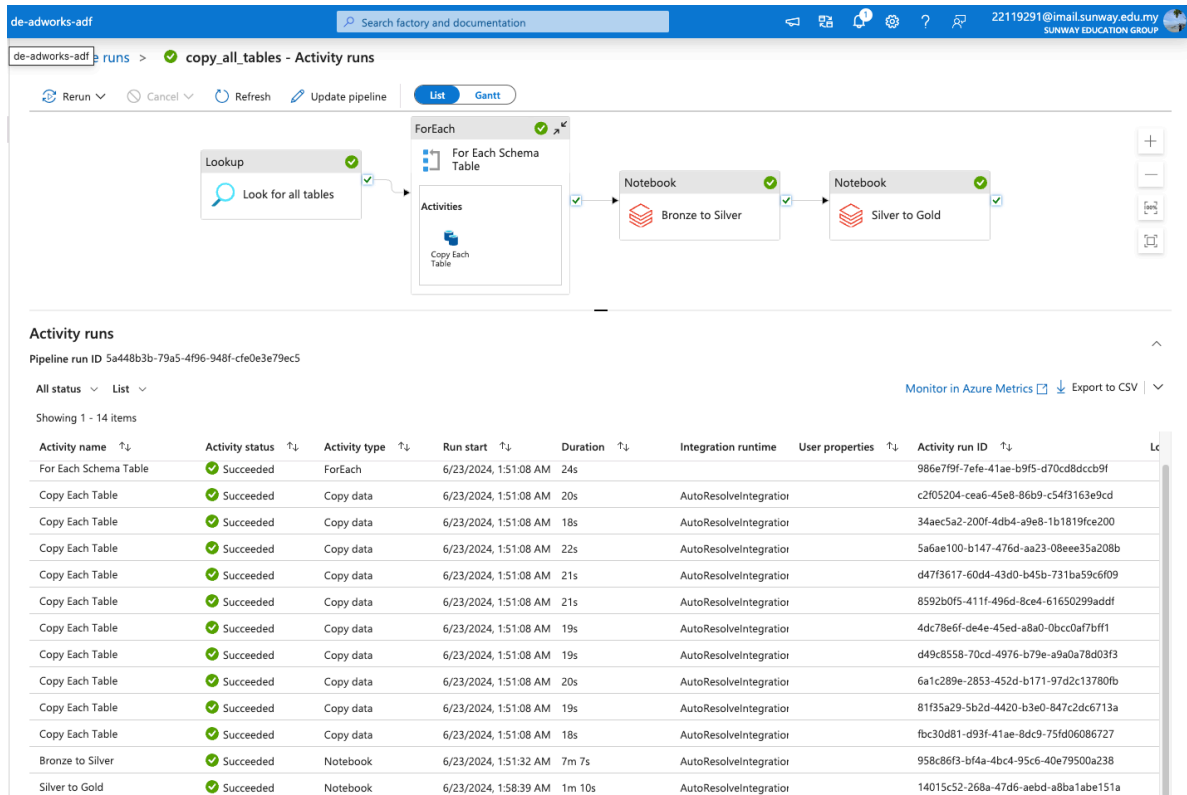
Pipeline runs

Triggered Debug Rerun Cancel options Refresh Edit columns List Gantt

Filter by run ID or name Kuala Lumpur, Singa... : Last 7 days Pipeline name: All Status: All Runs: Latest runs Triggered by: All Copy filters Export to CSV

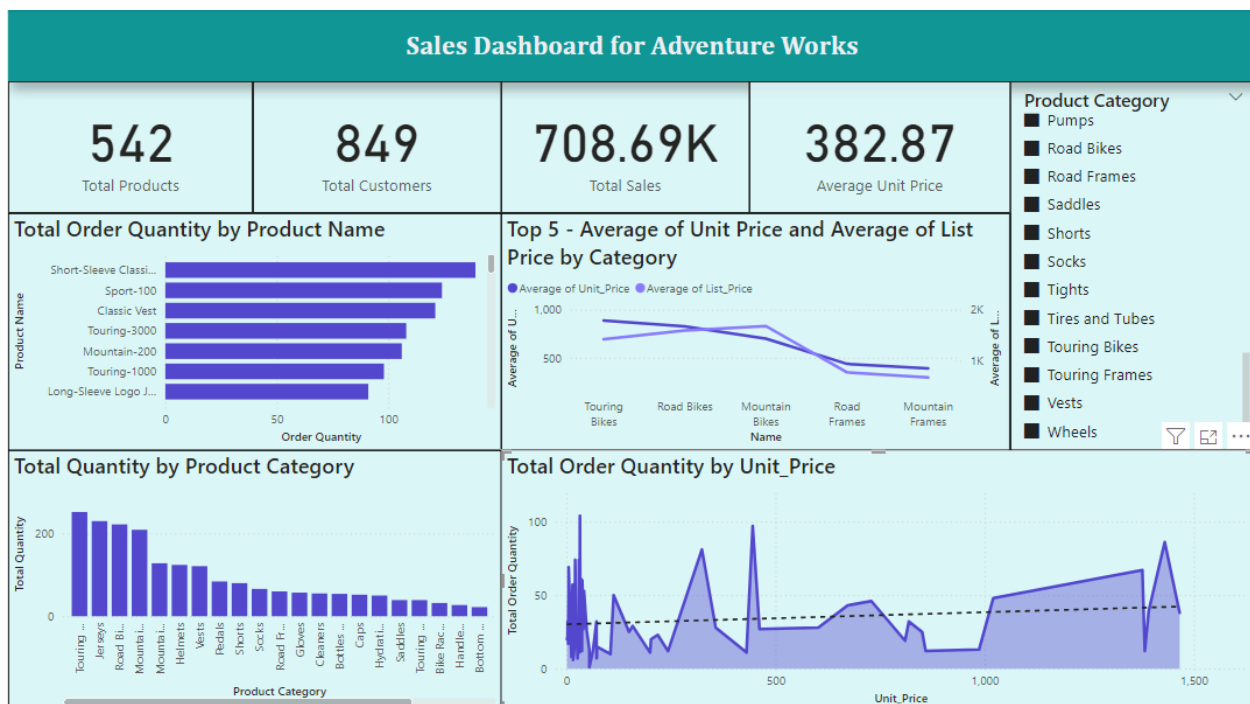
Showing 1 - 4 items Last refreshed 0 minutes ago

Pipeline name ↑↓	Run start ↑↓	Run end ↑↓	Duration	Triggered by	Status ↑↓	Run	Parameters	Annotations
copy_all_tables	6/23/2024, 1:51:00 AM	6/23/2024, 1:59:51 AM	8m 51s	scheduled_daily	✓ Succeeded	Original		
copy_all_tables	6/21/2024, 10:22:15 PM	6/21/2024, 10:30:51 PM	8m 36s	Manual trigger	✓ Succeeded	Original		
copy_all_tables	6/20/2024, 7:28:29 PM	6/20/2024, 7:29:01 PM	32s	Manual trigger	✓ Succeeded	Original		
copy_all_tables	6/20/2024, 7:25:47 PM	6/20/2024, 7:26:16 PM	30s	Manual trigger	✗ Failed	Original		



4. Validate result in Power BI dashboard

- The Total Customers has increased 2, from 847 to 849 as two new rows are added into the dataset.



Resource group:

Microsoft Azure

Search resources, services, and docs (G+/I)

22119291@mail.sunwa...
SUNWAY EDUCATION GROUP

Home > Resource groups >

de-adworks-rg

Resource group

Search

CreateManage viewDelete resource groupRefreshExport to CSVOpen queryAssign tagsMoveDeleteExport templateOpen in mobile

OverviewActivity logAccess control (IAM)TagsResource visualizerEventsSettingsDeploymentsSecurityDeployment stacksPoliciesPropertiesLocksCost ManagementCost analysisCost alerts (preview)Budgets

Essentials

Subscription (move) : [Azure for Students](#)Deployments : 2 Failed 6 Succeeded
Subscription ID : 9c4f9339-3621-4892-9643-bdb5ce85c0baLocation : East US
Tags (edit) : type : de

ResourcesRecommendations (1)

Filter for any field...Type equals allLocation equals allAdd filter

Showing 1 to 7 of 7 records.☐ Show hidden types

No groupingList view

Name	Type	Location
<input type="checkbox"/> Adventure-Works-d (de-adworks-server/Adventure-Works-d)	SQL database	Southeast Asia
<input type="checkbox"/> adworksadls	Storage account	Southeast Asia
<input type="checkbox"/> de-adworks-adf	Data factory (V2)	East US
<input type="checkbox"/> de-adworks-databricks	Azure Databricks Service	Southeast Asia
<input type="checkbox"/> de-adworks-kv	Key vault	East US
<input type="checkbox"/> de-adworks-server	SQL server	Southeast Asia
<input type="checkbox"/> de-adworks-synapse	Synapse workspace	Southeast Asia

Storage File Structure:

Microsoft Azure

Search resources, services, and docs (G+/I)

22119291@mail.sunwa...
SUNWAY EDUCATION GROUP

Home > Resource groups > de-adworks-rg > adworksadls

adworksadls | Containers

Storage account

Search

ContainerChange access levelRestore containersRefreshDeleteGive feedback

OverviewActivity logTagsDiagnose and solve problemsAccess Control (IAM)Data migrationEvents

Search containers by prefix

Show deleted containers

Name	Last modified	Anonymous access level	Lease state
<input type="checkbox"/> \$logs	6/20/2024, 4:12:37 PM	Private	Available
<input type="checkbox"/> bronze	6/20/2024, 4:15:35 PM	Private	Available
<input type="checkbox"/> gold	6/20/2024, 10:15:52 PM	Private	Available
<input type="checkbox"/> silver	6/20/2024, 10:15:47 PM	Private	Available

Azure Key Vault:

Microsoft Azure

Search resources, services, and docs (G+/I)

22119291@mail.sunwa...
SUNWAY EDUCATION GROUP

lworks-rg > de-adworks-kv

Secrets

Generate/ImportRefreshRestore BackupView sample codeManage deleted secrets

Name	Type	Status	Expiration date
dtbtoken		✓ Enabled	
password		✓ Enabled	
username		✓ Enabled	