Dokumentasi Data Pipelining Azure

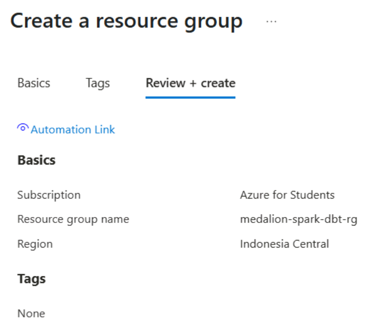
Dataset : Adventure Work

A. Configure Arch ADF, DBricks, Azure

1. Create and Config Architecture

a. Membuat Resource Group

Grup sumber daya dibuat sebagai wadah untuk semua komponen Azure yang akan digunakan.



Create a resource group ···

Basics | Tags | Review + create

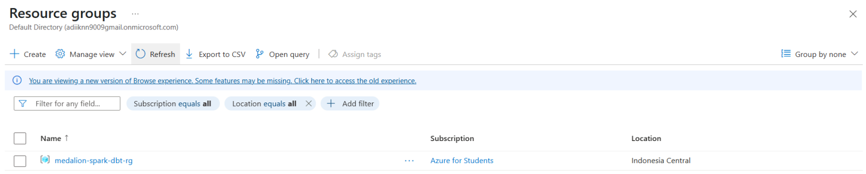
Automation Link

Basics

Subscription	Azure for Students
Resource group name	medalion-spark-dbt-rg
Region	Indonesia Central

Tags

None



Resource groups ···

Default Directory (adiiknn9009gmail.onmicrosoft.com)

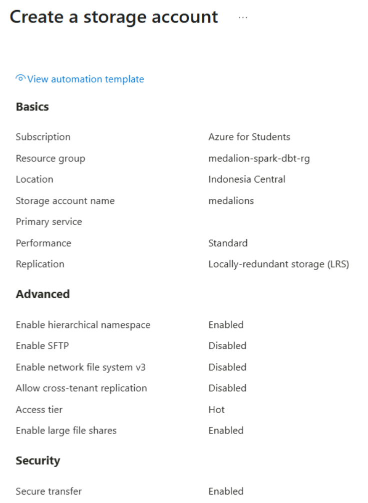
You are viewing a new version of Browse experience. Some features may be missing. Click here to access the old experience.

Filter for any field... | Subscription equals **all** | Location equals **all** | + Add filter

Name ↑	Subscription	Location
medalion-spark-dbt-rg	Azure for Students	Indonesia Central

b. Membuat Azure Data Lake Storage Gen2/ADLS Gen2

Dibuat sebagai penyimpanan utama untuk data.



Create a storage account ···

View automation template

Basics

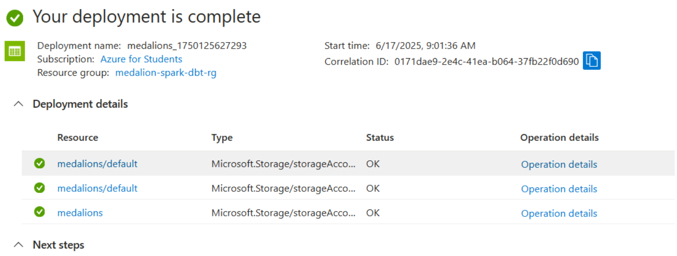
Subscription	Azure for Students
Resource group	medalion-spark-dbt-rg
Location	Indonesia Central
Storage account name	medalions
Primary service	
Performance	Standard
Replication	Locally-redundant storage (LRS)

Advanced

Enable hierarchical namespace	Enabled
Enable SFTP	Disabled
Enable network file system v3	Disabled
Allow cross-tenant replication	Disabled
Access tier	Hot
Enable large file shares	Enabled

Security

Secure transfer	Enabled



Your deployment is complete

Deployment name: medalions_1750125627293
Subscription: Azure for Students
Resource group: medalion-spark-dbt-rg

Start time: 6/17/2025, 9:01:36 AM
Correlation ID: 0171dae9-2e4c-41ea-b064-37fb22f0d690

Deployment details

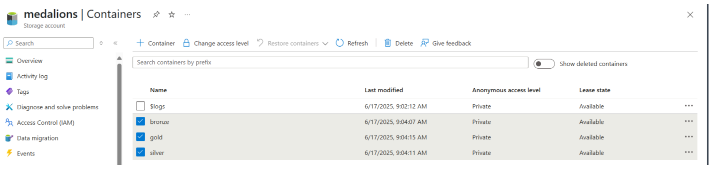
Resource	Type	Status	Operation details
medalions/default	Microsoft.Storage/storageAcco...	OK	Operation details
medalions/default	Microsoft.Storage/storageAcco...	OK	Operation details
medalions	Microsoft.Storage/storageAcco...	OK	Operation details

Next steps

c. Membuat Container for Bronze, Silver, Gold level

Container untuk Bronze, Silver, Gold level: Tiga container dibuat di ADLS Gen2 untuk menyimpan data pada berbagai tahap pemrosesan:

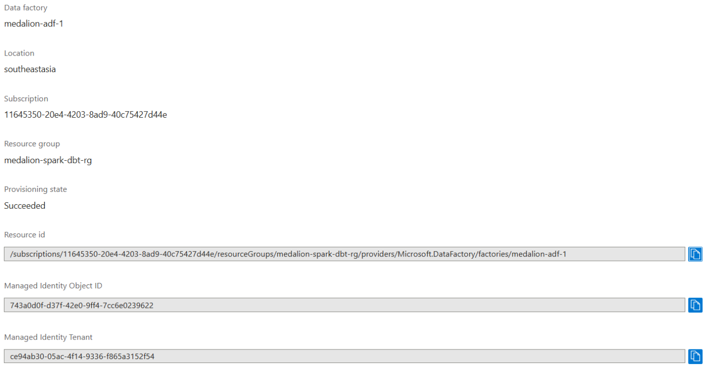
- Bronze: Data mentah dalam format aslinya
- Silver: Data yang telah dibersihkan dan diubah
- Gold: Data yang telah dimodelkan untuk analisis



	Name	Last modified	Anonymous access level	Lease state
□	$logs	6/17/2025, 9:02:12 AM	Private	Available
☑	bronze	6/17/2025, 9:04:07 AM	Private	Available
☑	gold	6/17/2025, 9:04:15 AM	Private	Available
☑	silver	6/17/2025, 9:04:11 AM	Private	Available

d. Membuat Data Factory

Dibuat sebagai orkestrator utama pipeline.



Data factory
medalion-adf-1

Location
southeastasia

Subscription
11645350-20e4-4203-8ad9-40c75427d44e

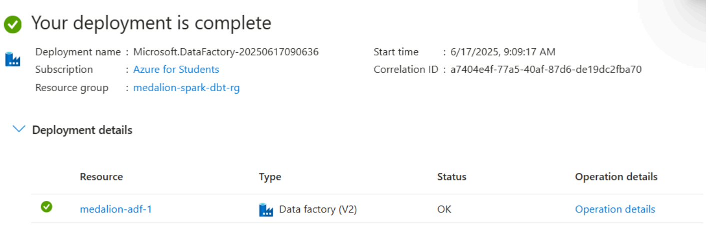
Resource group
medalion-spark-dbt-rg

Provisioning state
Succeeded

Resource id
/subscriptions/11645350-20e4-4203-8ad9-40c75427d44e/resourceGroups/medalion-spark-dbt-rg/providers/Microsoft.DataFactory/factories/medalion-adf-1

Managed Identity Object ID
743a0d0f-d37f-42e0-9ff4-7cc6e0239622

Managed Identity Tenant
ce94ab30-05ac-4f14-9336-f865a3152f54



Your deployment is complete

Deployment name : Microsoft.DataFactory-20250617090636
Subscription : Azure for Students
Resource group : medalion-spark-dbt-rg

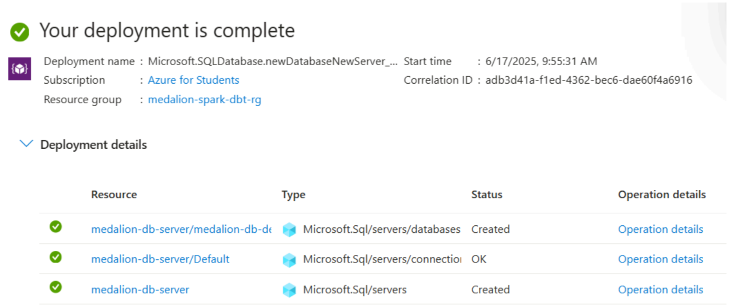
Start time : 6/17/2025, 9:09:17 AM
Correlation ID : a7404e4f-77a5-40af-87d6-de19dc2fba70

Deployment details

Resource	Type	Status	Operation details
medalion-adf-1	Data factory (V2)	OK	Operation details

e. Membuat SQL Database dan Server

Digunakan untuk menyimpan data sumber.



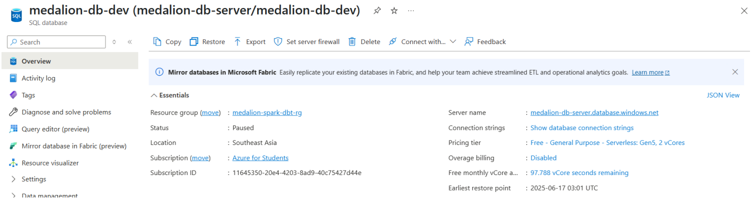
Your deployment is complete

Deployment name : Microsoft.SQLDatabase.newDatabaseNewServer_...
Subscription : Azure for Students
Resource group : medalion-spark-dbt-rg

Start time : 6/17/2025, 9:55:31 AM
Correlation ID : adb3d41a-f1ed-4362-bec6-dae60f4a6916

Deployment details

	Resource	Type	Status	Operation details
✓	medalion-db-server/medalion-db-dε	Microsoft.Sql/servers/databases	Created	Operation details
✓	medalion-db-server/Default	Microsoft.Sql/servers/connectioı	OK	Operation details
✓	medalion-db-server	Microsoft.Sql/servers	Created	Operation details



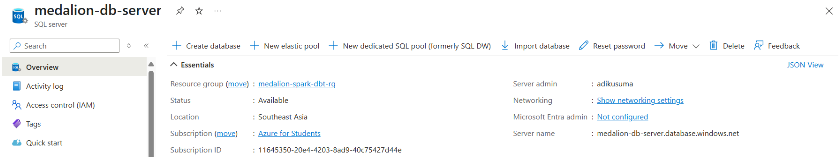
medalion-db-dev (medalion-db-server/medalion-db-dev)
SQL database

Mirror databases in Microsoft Fabric Easily replicate your existing databases in Fabric, and help your team achieve streamlined ETL and operational analytics goals. Learn more

Essentials

Resource group (move) : medalion-spark-dbt-rg
Status : Paused
Location : Southeast Asia
Subscription (move) : Azure for Students
Subscription ID : 11645350-20e4-4203-8ad9-40c75427d44e

Server name : medalion-db-server.database.windows.net
Connection strings : Show database connection strings
Pricing tier : Free - General Purpose - Serverless: Gen5, 2 vCores
Overage billing : Disabled
Free monthly vCore a... : 97.788 vCore seconds remaining
Earliest restore point : 2025-06-17 03:01 UTC



medalion-db-server
SQL server

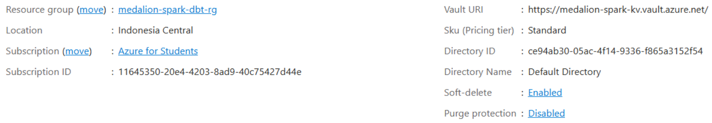
Essentials

Resource group (move) : medalion-spark-dbt-rg
Status : Available
Location : Southeast Asia
Subscription (move) : Azure for Students
Subscription ID : 11645350-20e4-4203-8ad9-40c75427d44e

Server admin : adikusuma
Networking : Show networking settings
Microsoft Entra admin : Not configured
Server name : medalion-db-server.database.windows.net

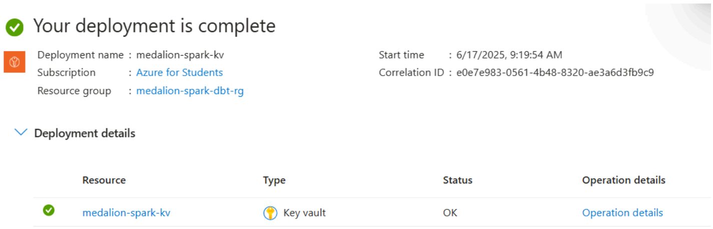
f. Membuat Key Vaults

Digunakan untuk menyimpan rahasia dan kredensial secara aman.



Resource group (move) : medalion-spark-dbt-rg
Location : Indonesia Central
Subscription (move) : Azure for Students
Subscription ID : 11645350-20e4-4203-8ad9-40c75427d44e

Vault URI : https://medalion-spark-kv.vault.azure.net/
Sku (Pricing tier) : Standard
Directory ID : ce94ab30-05ac-4f14-9336-f865a3152f54
Directory Name : Default Directory
Soft-delete : Enabled
Purge protection : Disabled



Your deployment is complete

Deployment name : medalion-spark-kv
Subscription : Azure for Students
Resource group : medalion-spark-dbt-rg

Start time : 6/17/2025, 9:19:54 AM
Correlation ID : e0e7e983-0561-4b48-8320-ae3a6d3fb9c9

Deployment details

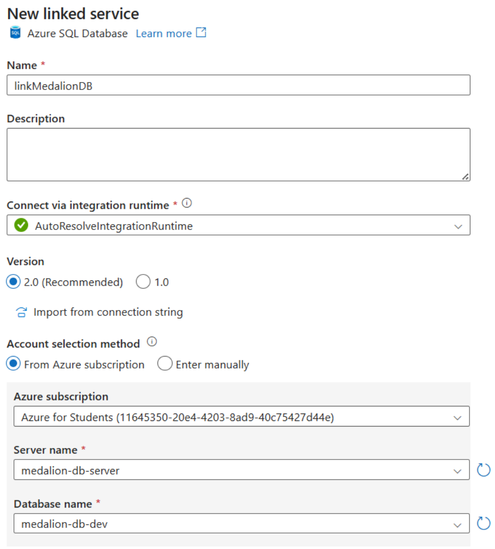
	Resource	Type	Status	Operation details
✓	medalion-spark-kv	Key vault	OK	Operation details

2. Linked on Data Factory

Data Factory dikonfigurasi untuk terhubung ke berbagai layanan:

a. Link SQL Database & Server

Link SQL Database & Server: Koneksi ke database sumber.



New linked service

Azure SQL Database Learn more

Name *

linkMedalionDB

Description

Connect via integration runtime *

AutoResolveIntegrationRuntime

Version

● 2.0 (Recommended) ○ 1.0

Import from connection string

Account selection method

● From Azure subscription ○ Enter manually

Azure subscription

Azure for Students (11645350-20e4-4203-8ad9-40c75427d44e)

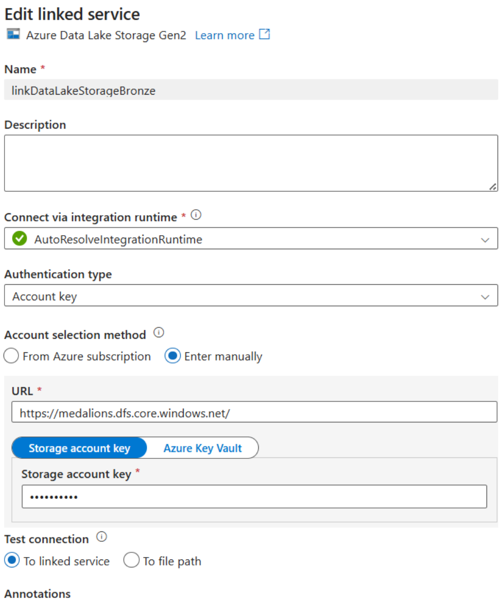
Server name *

medalion-db-server

Database name *

medalion-db-dev

b. Link Storage Account
Menambahkan linked ke storage account utama



Edit linked service

Azure Data Lake Storage Gen2 Learn more

Name *

linkDataLakeStorageBronze

Description

Connect via integration runtime *

AutoResolveIntegrationRuntime

Authentication type

Account key

Account selection method

○ From Azure subscription ● Enter manually

URL *

https://medalions.dfs.core.windows.net/

Storage account key | Azure Key Vault

Storage account key *

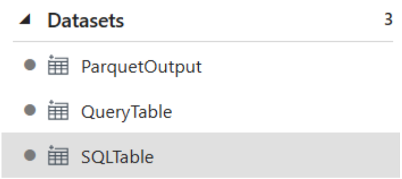
••••••••••

Test connection

● To linked service ○ To file path

Annotations

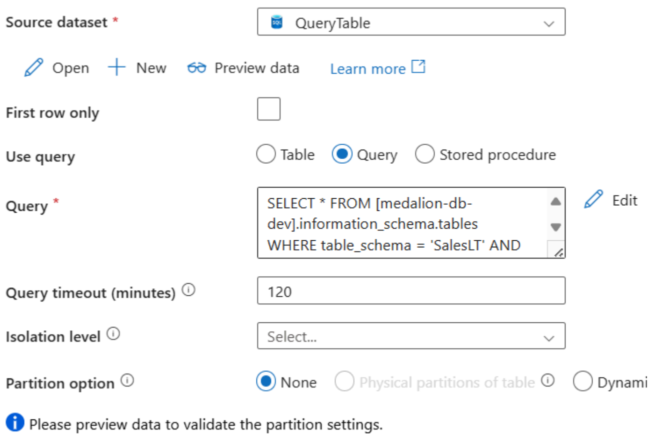
3. Pipeline Activity/Process on Data Factory
   Membuat dataset antara lain :
   a. Parquetoutput : digunakan untuk menerima data hasil dari for each copy data yang akan distore kan pada storage account
   b. Query Table : memuata data set yang akan digunakan berdasarkan query yang digunakan disini menggunakan untuk memanggil semua table schema SalesST dari table_type
   c. SQL Table : memuat data set yang akan digunakan sebagai copy dari for each yang akan menerima iterasi sebanyak hasil dari LOOKUP query sebelumnya
   d. Create Dataset



◢ Datasets 3

- ParquetOutput
- QueryTable
- SQLTable

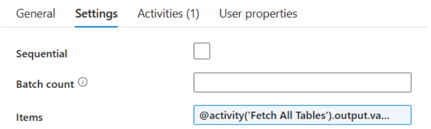
e. Create LOOKUP Activity on Pipeline
   - Menggunakan QueryTable sebagai sumber
   - Query: SELECT * FROM [predation-db-dev].information_schema.tables WHERE table_schema = "SelectT" AND table_name = “Base Table”
   - Digunakan untuk mendapatkan daftar tabel yang akan diproses



Source dataset *	QueryTable
First row only	□
Use query	○ Table ● Query ○ Stored procedure
Query *	SELECT * FROM [medalion-db-dev].information_schema.tables WHERE table_schema = 'SalesLT' AND
Query timeout (minutes)	120
Isolation level	Select...
Partition option	● None ○ Physical partitions of table ○ Dynami

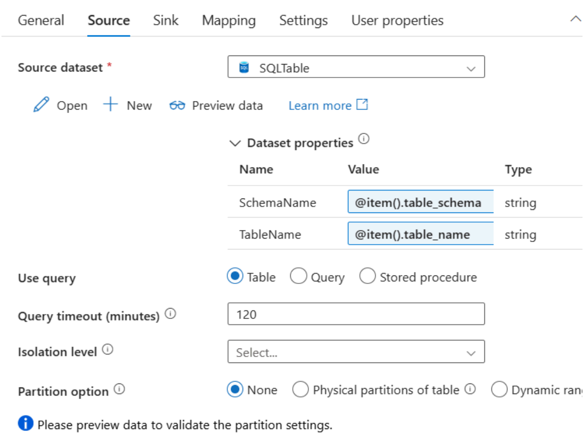
Please preview data to validate the partition settings.

f. Create FOREACH Activity on Pipeline
   - Memproses setiap table yang ditemukan oleh LOOKUP
   - Menjalankan iterasi secara sequential



Sequential	□
Batch count	
Items	@activity('Fetch All Tables').output.va...

g. Create Copy Data Activity on FOREACH Activity
   Di dalam FOREACH, menyalin data dari sumber ke tujuan, dengan tujuan hasil akan disimpan pada container bronze yang digunakan untuk mengmount data raw



Source dataset * SQLTable

Dataset properties

Name	Value	Type
SchemaName	@item().table_schema	string
TableName	@item().table_name	string

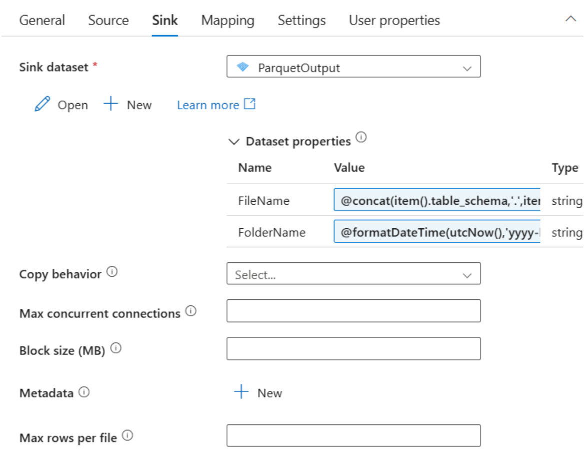
Use query ● Table ○ Query ○ Stored procedure

Query timeout (minutes) 120

Isolation level Select...

Partition option ● None ○ Physical partitions of table ○ Dynamic ran

Please preview data to validate the partition settings.



Sink dataset * ParquetOutput

Dataset properties

Name	Value	Type
FileName	@concat(item().table_schema,'.',iter	string
FolderName	@formatDateTime(utcNow(),'yyyy-I	string

Copy behavior Select...

Max concurrent connections

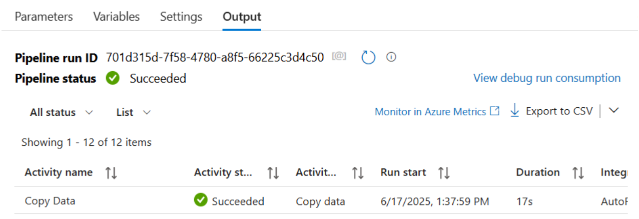
Block size (MB)

Metadata

Max rows per file

h. Debug Run Pipeline

Melakukan debug untuk menmountkan data set yang dilakukan pada pipeline ke dalam container bronze pada storage account dengan tipe data .parquet yang akan digunakan nantinya untuk dilakukan cleaning dan monitoring

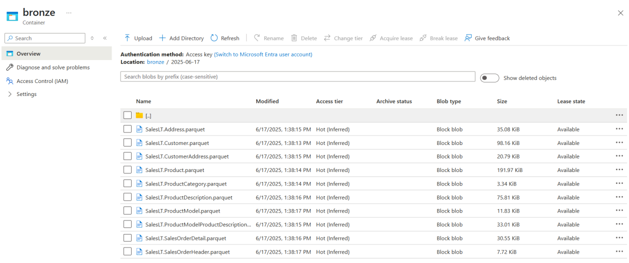


Pipeline run ID 701d315d-7f58-4780-a8f5-66225c3d4c50

Pipeline status ✓ Succeeded

Showing 1 - 12 of 12 items

Activity name	Activity st...	Activit...	Run start	Duration	Integ
Copy Data	✓ Succeeded	Copy data	6/17/2025, 1:37:59 PM	17s	AutoF



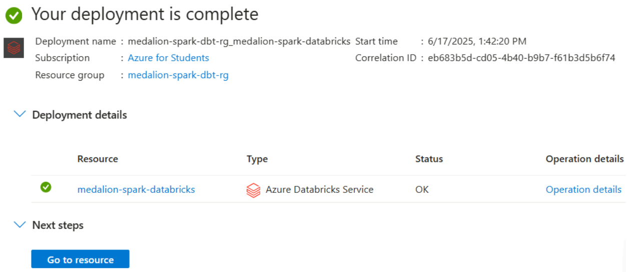
Authentication method: Access key (Switch to Microsoft Entra user account)
Location: bronze / 2025-06-17

Search blobs by prefix (case-sensitive)

Show deleted objects

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
[..]						
SalesLT.Address.parquet	6/17/2025, 1:38:15 PM	Hot (Inferred)		Block blob	35.08 KiB	Available
SalesLT.Customer.parquet	6/17/2025, 1:38:13 PM	Hot (Inferred)		Block blob	98.16 KiB	Available
SalesLT.CustomerAddress.parquet	6/17/2025, 1:38:15 PM	Hot (Inferred)		Block blob	20.79 KiB	Available
SalesLT.Product.parquet	6/17/2025, 1:38:14 PM	Hot (Inferred)		Block blob	191.97 KiB	Available
SalesLT.ProductCategory.parquet	6/17/2025, 1:38:14 PM	Hot (Inferred)		Block blob	3.34 KiB	Available
SalesLT.ProductDescription.parquet	6/17/2025, 1:38:16 PM	Hot (Inferred)		Block blob	75.81 KiB	Available
SalesLT.ProductModel.parquet	6/17/2025, 1:38:17 PM	Hot (Inferred)		Block blob	11.83 KiB	Available
SalesLT.ProductModelProductDescription...	6/17/2025, 1:38:15 PM	Hot (Inferred)		Block blob	33.01 KiB	Available
SalesLT.SalesOrderDetail.parquet	6/17/2025, 1:38:16 PM	Hot (Inferred)		Block blob	30.55 KiB	Available
SalesLT.SalesOrderHeader.parquet	6/17/2025, 1:38:17 PM	Hot (Inferred)		Block blob	7.72 KiB	Available

5. Create Data Bricks



Your deployment is complete

Deployment name : medalion-spark-dbt-rg_medalion-spark-databricks
Subscription : Azure for Students
Resource group : medalion-spark-dbt-rg
Start time : 6/17/2025, 1:42:20 PM
Correlation ID : eb683b5d-cd05-4b40-b9b7-f61b3d5b6f74

Deployment details

	Resource	Type	Status	Operation details
✓	medalion-spark-databricks	Azure Databricks Service	OK	Operation details

Next steps

6. Menghubungkan databricks dengan storage account untuk masing masing container : bronze, silver, gold :

Menghubungkan Databricks ke ADLS Gen2 untuk akses data di container bronze/silver/gold.

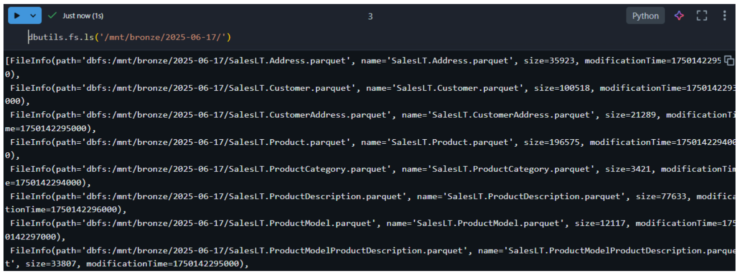
```
dbutils.fs.mount(
    source="wasbs://bronze@medalions.blob.core.windows.net",
    mount_point="/mnt/bronze",

extra_configs={"fs.azure.account.key.medalions.blob.core.windo
ws.net":
dbutils.secrets.get('DataBricksScope','storageAccountKey')}
)

dbutils.fs.mount(
    source="wasbs://silver@medalions.blob.core.windows.net",
    mount_point="/mnt/silver",

extra_configs={"fs.azure.account.key.medalions.blob.core.windo
ws.net":
dbutils.secrets.get('DataBricksScope','storageAccountKey')}
)

dbutils.fs.mount(
    source="wasbs://gold@medalions.blob.core.windows.net",
    mount_point="/mnt/gold",

extra_configs={"fs.azure.account.key.medalions.blob.core.windo
ws.net":
dbutils.secrets.get('DataBricksScope','storageAccountKey')}
)
```

- source: Path ke container Azure Blob Storage (wasbs://[container]@[storage-account].blob.core.windows.net).
- mount_point: Direktori virtual di Databricks (/mnt/bronze).

- extra_configs: Mengambil kunci akses storage dari Azure Key Vault (DataBricksScope).

```
dbutils.fs.ls('/mnt/bronze/2025-06-17/')
```

Output Success :



```
dbutils.fs.ls('/mnt/bronze/2025-06-17/')
```

```
[FileInfo(path='dbfs:/mnt/bronze/2025-06-17/SalesLT.Address.parquet', name='SalesLT.Address.parquet', size=35923, modificationTime=1750142295
0),
 FileInfo(path='dbfs:/mnt/bronze/2025-06-17/SalesLT.Customer.parquet', name='SalesLT.Customer.parquet', size=100518, modificationTime=175014229
000),
 FileInfo(path='dbfs:/mnt/bronze/2025-06-17/SalesLT.CustomerAddress.parquet', name='SalesLT.CustomerAddress.parquet', size=21289, modificationTi
me=1750142295000),
 FileInfo(path='dbfs:/mnt/bronze/2025-06-17/SalesLT.Product.parquet', name='SalesLT.Product.parquet', size=196575, modificationTime=175014229400
0),
 FileInfo(path='dbfs:/mnt/bronze/2025-06-17/SalesLT.ProductCategory.parquet', name='SalesLT.ProductCategory.parquet', size=3421, modificationTim
e=1750142294000),
 FileInfo(path='dbfs:/mnt/bronze/2025-06-17/SalesLT.ProductDescription.parquet', name='SalesLT.ProductDescription.parquet', size=77633, modifica
tionTime=1750142296000),
 FileInfo(path='dbfs:/mnt/bronze/2025-06-17/SalesLT.ProductModel.parquet', name='SalesLT.ProductModel.parquet', size=12117, modificationTime=175
0142297000),
 FileInfo(path='dbfs:/mnt/bronze/2025-06-17/SalesLT.ProductModelProductDescription.parquet', name='SalesLT.ProductModelProductDescription.parque
t', size=33807, modificationTime=1750142295000),
```

Penjelasan : Container bronze/silver/gold bisa diakses via /mnt/[level] di Databricks.

BaseNotebook

```
folder_name = dbutils.widgets.get('folder_name')
table_schema = dbutils.widgets.get('table_schema')
table_name = dbutils.widgets.get('table_name')

# Buat database jika belum ada
spark.sql(f"CREATE DATABASE IF NOT EXISTS `{table_schema}`")

# Buat tabel jika belum ada
spark.sql(f"""
        CREATE TABLE IF NOT EXISTS
`{table_schema}`.`{table_name}`
        USING PARQUET
        LOCATION
'/mnt/bronze/{folder_name}/{table_schema}.{table_name}.parquet
'
        """)
```

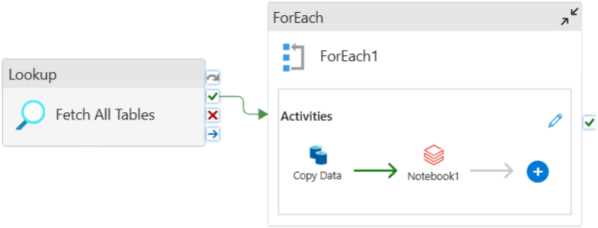
Penjelasan :

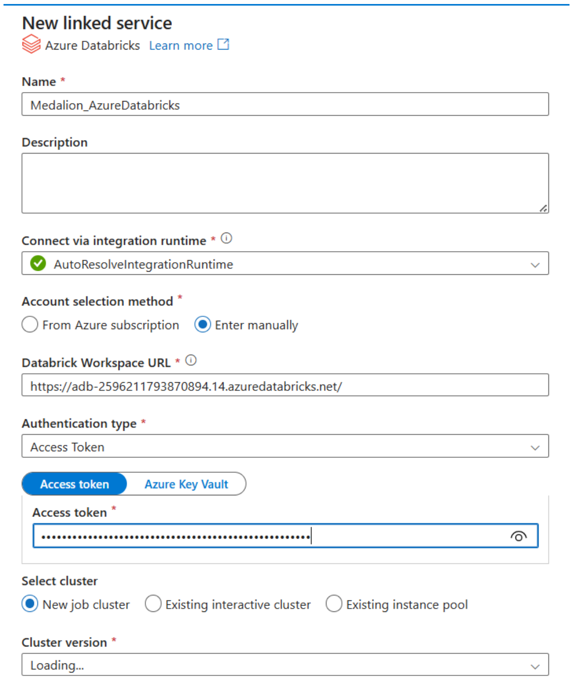
Membuat tabel di Databricks dari file Parquet di ADLS.

- Parameter Di-pass dari ADF (e.g., folder_name=2025-06-17).
- LOCATION: Path ke file Parquet di ADLS.
- USING PARQUET: Format file sumber.

Output : Tabel Databricks (e.g., saleslt.address) yang linked ke file Parquet di ADLS.

7. Menambahkan Notebook / Data Bricks pada pipeline FOREACH
    a. Konfigurasi Data Bricks





New linked service

Azure Databricks Learn more

Name *

Medalion_AzureDatabricks

Description

Connect via integration runtime *

AutoResolveIntegrationRuntime

Account selection method *

○ From Azure subscription ● Enter manually

Databrick Workspace URL *

https://adb-2596211793870894.14.azuredatabricks.net/

Authentication type *

Access Token

Access token | Azure Key Vault

Access token *

••

Select cluster

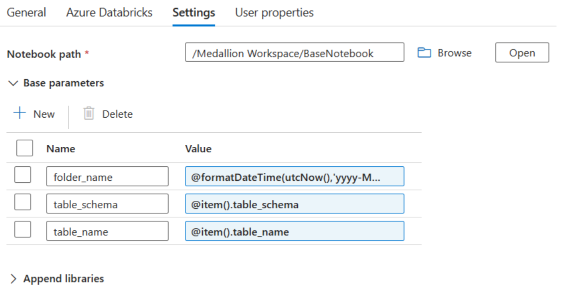
● New job cluster ○ Existing interactive cluster ○ Existing instance pool

Cluster version *

Loading...

Penjelasna : menambahkan proses atau service pada pipeline didalam foreach untuk digunakan setelah data berhail di mount pada pipeline ketika copy maka selanjutna akan di mountkan ke dalam bronze container menggunakan databricks yang dihubngkan ke dalam network untuk dilakukan proses mountingnya untuk datanya agar dapat domount pada workload id databrick

b. Setting Params & Notebook Path



Notebook path * /Medallion Workspace/BaseNotebook

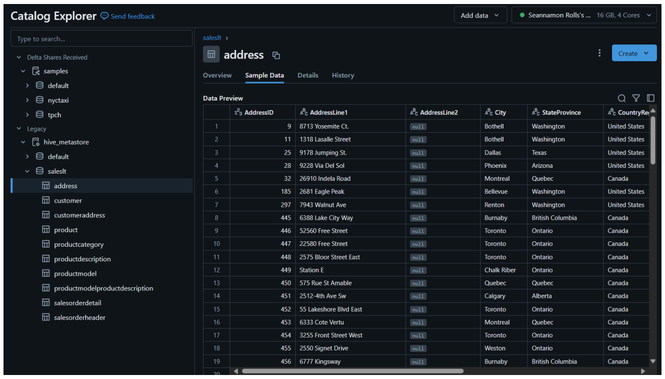
Base parameters

	Name	Value
□	folder_name	@formatDateTime(utcNow(),'yyyy-M...
□	table_schema	@item().table_schema
□	table_name	@item().table_name

Append libraries

Penjelasan : Menambahkan konfigurasi parameter yang akan diload pada basenotebook atau db notebook yang dijalankan dimana akan melakukan check juga untuk data yang tidak tertera untuk table_schema mapun table_name nya.

c. Testing



	AddressID	AddressLine1	AddressLine2	City	StateProvince	CountryRe
1	9	8713 Yosemite Ct.	null	Bothell	Washington	United States
2	11	1318 Lasalle Street	null	Bothell	Washington	United States
3	25	9178 Jumping St.	null	Dallas	Texas	United States
4	28	9228 Via Del Sol	null	Phoenix	Arizona	United States
5	32	26910 Indela Road	null	Montreal	Quebec	Canada
6	185	2681 Eagle Peak	null	Bellevue	Washington	United States
7	297	7943 Walnut Ave	null	Renton	Washington	United States
8	445	6388 Lake City Way	null	Burnaby	British Columbia	Canada
9	446	52560 Free Street	null	Toronto	Ontario	Canada
10	447	22580 Free Street	null	Toronto	Ontario	Canada
11	448	2575 Bloor Street East	null	Toronto	Ontario	Canada
12	449	Station E	null	Chalk Riber	Ontario	Canada
13	450	575 Rue St Amable	null	Quebec	Quebec	Canada
14	451	2512-4th Ave Sw	null	Calgary	Alberta	Canada
15	452	55 Lakeshore Blvd East	null	Toronto	Ontario	Canada
16	453	6333 Cote Vertu	null	Montreal	Quebec	Canada
17	454	3255 Front Street West	null	Toronto	Ontario	Canada
18	455	2550 Signet Drive	null	Weston	Ontario	Canada
19	456	6777 Kingsway	null	Burnaby	British Columbia	Canada

Penjelasan :

Hasil dari data yang di mountkan pada copy yang digunakan pada notebook untuk diload pada workload database pada databrick azure berhasil dibuat sama dengan isi dataset pada bronze level

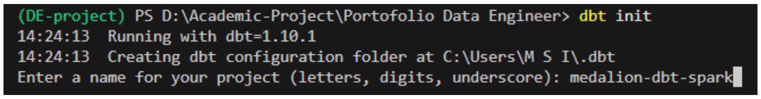
2. DBT (Data Build Tool) Transformation

a. DBT Configuration

i. Installation library :

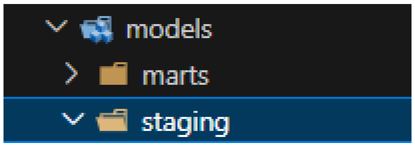
- Dbt-databricks = 1.10.1
- databricks cli = 1.10.3

b. Initialization dbt



```
(DE-project) PS D:\Academic-Project\Portofolio Data Engineer> dbt init
14:24:13  Running with dbt=1.10.1
14:24:13  Creating dbt configuration folder at C:\Users\M S I\.dbt
Enter a name for your project (letters, digits, underscore): medalion-dbt-spark
```

c. Creating file



bronze.yml

```
version: 2

sources:
  - name: saleslt
    schema: saleslt
    description: This is the adventureworks database loaded
into bronze
    tables:
      - name: address
      - name: customer
      - name: customeraddress
      - name: product
      - name: productcategory
      - name: productdescription
      - name: productmodel
      - name: salesorderdetail
      - name: salesorderheader
```

Penjelasan :

Deklarasi Sumber Data: Daftar tabel mentah yang akan digunakan di seluruh proyek DBT. Mengintegrasikan DBT dengan data yang sudah ada di load pada bronze layer, dan membuat metadata yang bisa dilacak DBT docs

d. SQL Silver Level

Membuat snapshot historis data dari sumber ke silver layer.

i. Creating address.sql

address.sql

```
{% snapshot address_snapshot %}

{{
    config(
      file_format = "delta",
      location_root = "/mnt/silver/address",
      target_schema='snapshots',
      invalidate_hard_deletes=True,
      unique_key='AddressID',
      strategy='check',
      check_cols='all'
    )
}}

with source_data as (
    select
        AddressID,
        AddressLine1,
        AddressLine2,
        City,
        StateProvince,
        CountryRegion,
        PostalCode
    from {{ source('saleslt', 'address') }}
)
select *
from source_data

{% endsnapshot %}
```

Penjelasan :

- File_format : Delta Lake (format yang dipilih untuk silver).
- Unique_key : Kolom PK untuk tracking perubahan
- strategy='check': Bandingkan nilai kolom (check_cols='all') untuk deteksi perubahan.
- location_root: Penyimpanan fisik file Delta.

Output

Tabel snapshot (e.g., snapshots.address_snapshot) dengan kolom tambahan:

- dbt_valid_from: Timestamp awal validitas record.
- dbt_valid_to: Timestamp akhir validitas (NULL untuk data terbaru).

ii. Creating customer.sql

customer.sql

```
{% snapshot customer_snapshot %}

{{
    config(
      file_format = "delta",
      location_root = "/mnt/silver/customer",

      target_schema='snapshots',
      invalidate_hard_deletes=True,
      unique_key='CustomerId',
      strategy='check',
      check_cols='all'
    )
}}

with source_data as (
    select
        CustomerId,
        NameStyle,
        Title,
        FirstName,
        MiddleName,
        LastName,
        Suffix,
        CompanyName,
        SalesPerson,
        EmailAddress,
        Phone,
        PasswordHash,
        PasswordSalt
    from {{ source('saleslt', 'customer') }}
)
select *
from source_data
```

```
{% endsnapshot %}
```

iii. Creating customeraddress.sql

customeraddress.sql

```
{% snapshot customeraddress_snapshot %}

{{
    config(
      file_format = "delta",
      location_root = "/mnt/silver/customeraddress",

      target_schema='snapshots',
      invalidate_hard_deletes=True,
      unique_key="CustomerId||'-'||AddressId",
      strategy='check',
      check_cols='all'
    )
}}

with source_data as (
    select
        CustomerId,
        AddressId,
        AddressType
    from {{ source('saleslt', 'customeraddress') }}
)
select *
from source_data

{% endsnapshot %}
```

iv. Creating product.sql

```
product.sql
```

```
{% snapshot product_snapshot %}

{{
    config(
      file_format = "delta",
      location_root = "/mnt/silver/product",

      target_schema='snapshots',
      invalidate_hard_deletes=True,
      unique_key='ProductID',
      strategy='check',
      check_cols='all'
    )
}}

with product_snapshot as (
    SELECT
        ProductID,
        Name,
        ProductNumber,
        Color,
        StandardCost,
        ListPrice,
        Size,
        Weight,
        ProductCategoryID,
        ProductModelID,
        SellStartDate,
        SellEndDate,
        DiscontinuedDate,
        ThumbNailPhoto,
        ThumbnailPhotoFileName
    FROM {{ source('saleslt', 'product') }}
)

select * from product_snapshot

{% endsnapshot %}
```

v. Creating productmodel.sql

productmodel.sql

```
{% snapshot productmodel_snapshot %}

{{
    config(
      file_format = "delta",
      location_root = "/mnt/silver/productmodel",
      target_schema='snapshots',
      invalidate_hard_deletes=True,
      unique_key='ProductModelID',
      strategy='check',
      check_cols='all'
    )
}}

with product_snapshot as (
    SELECT
        ProductModelID,
        Name,
        CatalogDescription
    FROM {{ source('saleslt', 'productmodel') }}
)

select * from product_snapshot

{% endsnapshot %}
```

vi. Creating salesorderdetail.sql

salesorderdetail.sql

```
{% snapshot salesorderdetail_snapshot %}

{{
    config(
      file_format = "delta",
      location_root = "/mnt/silver/salesorderdetail",
      target_schema='snapshots',
      invalidate_hard_deletes=True,
      unique_key='SalesOrderDetailID',
      strategy='check',
      check_cols='all'
    )
}}

with salesorderdetail_snapshot as (
    SELECT
        SalesOrderID,
        SalesOrderDetailID,
        OrderQty,
        ProductID,
        UnitPrice,
        UnitPriceDiscount,
        LineTotal
    FROM {{ source('saleslt', 'salesorderdetail') }}
)

select * from salesorderdetail_snapshot

{% endsnapshot %}
```

vii. Creating salesorderheader.sql

salesorderheader.sql

```
{% snapshot salesorderheader_snapshot %}

{{
    config(
      file_format = "delta",
      location_root = "/mnt/silver/salesorderheader",
      target_schema='snapshots',
      invalidate_hard_deletes=True,
      unique_key='SalesOrderID',
      strategy='check',
      check_cols='all'
    )
}}

with salesorderheader_snapshot as (
    SELECT
        SalesOrderID,
        RevisionNumber,
        OrderDate,
        DueDate,
        ShipDate,
        Status,
        OnlineOrderFlag,
        SalesOrderNumber,
        PurchaseOrderNumber,
        AccountNumber,
        CustomerID,
        ShipToAddressID,
        BillToAddressID,
        ShipMethod,
        CreditCardApprovalCode,
        SubTotal,
        TaxAmt,
        Freight,
        TotalDue,
        Comment
    FROM {{ source('saleslt', 'salesorderheader') }}
)

select * from salesorderheader_snapshot

{% endsnapshot %}
```

e. GOLDEN SQL level (marts)
   Transformasi data silver ke dimensional model (star schema).

   i. Customer
      Creating dim_customer.sql

dim_customer.sql

```
{{
    config(
        materialized = "table",
        file_format = "delta",
        location_root = "/mnt/gold/customers"
    )
}}

with address_snapshot as (
    select
        AddressID,
        AddressLine1,
        AddressLine2,
        City,
        StateProvince,
        CountryRegion,
        PostalCode
    from {{ ref('address_snapshot') }} where dbt_valid_to
is null
)

, customeraddress_snapshot as (
    select
        CustomerId,
        AddressId,
        AddressType
    from {{ref('customeraddress_snapshot')}} where
dbt_valid_to is null
)

, customer_snapshot as (
    select
        CustomerId,
        concat(ifnull(FirstName,' '),'
',ifnull(MiddleName,' '),' ',ifnull(LastName,' ')) as
FullName
    from {{ref('customer_snapshot')}} where dbt_valid_to
is null
)

, transformed as (
    select
    row_number() over (order by
customer_snapshot.customerid) as customer_sk, --
auto-incremental surrogate key
    customer_snapshot.CustomerId,
    customer_snapshot.fullname,
    customeraddress_snapshot.AddressID,
    customeraddress_snapshot.AddressType,
    address_snapshot.AddressLine1,
    address_snapshot.City,
    address_snapshot.StateProvince,
    address_snapshot.CountryRegion,
    address_snapshot.PostalCode
    from customer_snapshot
    inner join customeraddress_snapshot on
customer_snapshot.CustomerId =
customeraddress_snapshot.CustomerId
    inner join address_snapshot on
customeraddress_snapshot.AddressID =
address_snapshot.AddressID
)
select *
from transformed
```

Penjelasan :

- materialized = "table": Buat tabel fisik (bukan view).
- location_root: Path penyimpanan di gold layer.
- ref(): Referensi ke model DBT lain (e.g., address_snapshot).
- row_number() as customer_sk: Generate surrogate key.

Output:

- Tabel dimensi (dim_customers) di /mnt/gold/customers dengan schema:

Creating dim_customer.yml

dim_customer.yml

```
version: 2

models:
  - name: dim_customers
    columns:
      - name: customer_sk
        description: The surrogate key of the customer
        tests:
          - unique
          - not_null
      - name: customerid
        description: The natural key of the customer
        tests:
          - not_null
      - name: fullname
        description: The customer name. Adopted as
customer_fullname when person name is not null.
      - name: AddressId
        tests:
          - not_null
      - name: AddressType
      - name: AddressLine1
      - name: City
      - name: StateProvince
      - name: CountryRegion
      - name: PostalCode
```

ii. Product

Creating dim_product.sql

dim_product.sql

```
{{
    config(
        materialized = "table",
        file_format = "delta",
        location_root = "/mnt/gold/products"
    )
}}

with product_snapshot as (
    select
        productId,
        name,
        standardCost,
        listPrice,
        size,
        weight,
        productcategoryid,
        productmodelid,
        sellstartdate,
        sellenddate,
        discontinueddate
    from {{ ref("product_snapshot") }}
    where dbt_valid_to is null
),

product_model_snapshot as (
    select
        productmodelid,
        name,
        CatalogDescription,
        row_number() over (order by name) as model_id
    from {{ ref("productmodel_snapshot") }}
    where dbt_valid_to is null
),


transformed as (
    select
        row_number() over (order by p.productId) as
product_sk,
        p.name as product_name,
        p.standardCost,
        p.listPrice,
        p.size,
        p.weight,
        pm.name as model,
        pm.CatalogDescription as description,
        p.sellstartdate,
        p.sellenddate,
        p.discontinueddate
    from product_snapshot p
    left join product_model_snapshot pm on
p.productmodelid = pm.productmodelid
)

select * from transformed
```

Creating dim_customer.yml

dim_product.yml

```
version: 2

models:
  - name: dim_products
    columns:
      - name: product_sk
        description: The surrogate key of the product
        tests:
          - unique
          - not_null
      - name: product_name
        description: The name of the product
        tests:
          - not_null
      - name: standard_cost
        description: The standard cost of the product
      - name: list_price
        description: The list price of the product
      - name: size
        description: The size of the product
      - name: weight
        description: The weight of the product
      - name: category
        description: The category of the product
      - name: model
        description: The model of the product
      - name: description
        description: The description of the product
      - name: sellstartdate
        description: The date when the product is
available for sale
        tests:
          - not_null
      - name: sellenddate
        description: The date when the product is no longer available for sale
      - name: discontinueddate
        description: The date when the product is discontinued
```

iii. Sales

Creating dim_customer.sql

sales.sql

```
{{
    config(
        materialized = "table",
        file_format = "delta",
        location_root = "/mnt/gold/sales"
    )
}}

with salesorderdetail_snapshot as (
    SELECT
        SalesOrderID,
        SalesOrderDetailID,
        OrderQty,
        ProductID,
        UnitPrice,
        UnitPriceDiscount,
        LineTotal
    FROM {{ ref("salesorderdetail_snapshot") }}
),

product_snapshot as (
    SELECT
        ProductID,
        Name,
        ProductNumber,
        Color,
        StandardCost,
        ListPrice,
        Size,
        Weight,
        SellStartDate,
        SellEndDate,
        DiscontinuedDate,
        ThumbNailPhoto,
        ThumbnailPhotoFileName
    FROM {{ source('saleslt', 'product') }}
),
saleorderheader_snapshot as (
    SELECT
        SalesOrderID,
        RevisionNumber,
        OrderDate,
        DueDate,
        ShipDate,
        Status,
        OnlineOrderFlag,
        SalesOrderNumber,
        PurchaseOrderNumber,
        AccountNumber,
        CustomerID,
        ShipToAddressID,
        BillToAddressID,
        ShipMethod,
        CreditCardApprovalCode,
        SubTotal,
        TaxAmt,
        Freight,
        TotalDue,
        Comment,
        row_number() over (partition by SalesOrderID
order by SalesOrderID) as row_num
    FROM {{ source('saleslt', 'salesorderheader') }}
),

transformed as (
    select
        sod.SalesOrderID,
        sod.SalesOrderDetailID,
        sod.OrderQty,
        sod.ProductID,
        sod.UnitPrice,
        sod.UnitPriceDiscount,
        sod.LineTotal,
        p.Name,
        p.ProductNumber,
        p.Color,
        p.StandardCost,
        p.ListPrice,
        p.Size,
        p.Weight,
        p.SellStartDate,
        p.SellEndDate,
        p.DiscontinuedDate,
        p.ThumbNailPhoto,
        p.ThumbnailPhotoFileName,
        soh.RevisionNumber,
        soh.OrderDate,
        soh.DueDate,
        soh.ShipDate,
        soh.Status,
        soh.OnlineOrderFlag,
        soh.SalesOrderNumber,
        soh.PurchaseOrderNumber,
        soh.AccountNumber,
        soh.CustomerID,
        soh.ShipToAddressID,
        soh.BillToAddressID,
        soh.ShipMethod,
        soh.CreditCardApprovalCode,
        soh.SubTotal,
        soh.TaxAmt,
        soh.Freight,
        soh.TotalDue,
        soh.Comment
    from salesorderdetail_snapshot sod
    left join product_snapshot p on sod.ProductID =
p.ProductID
    left join saleorderheader_snapshot soh on
sod.SalesOrderID = soh.SalesOrderID
)

select * from transformed
```

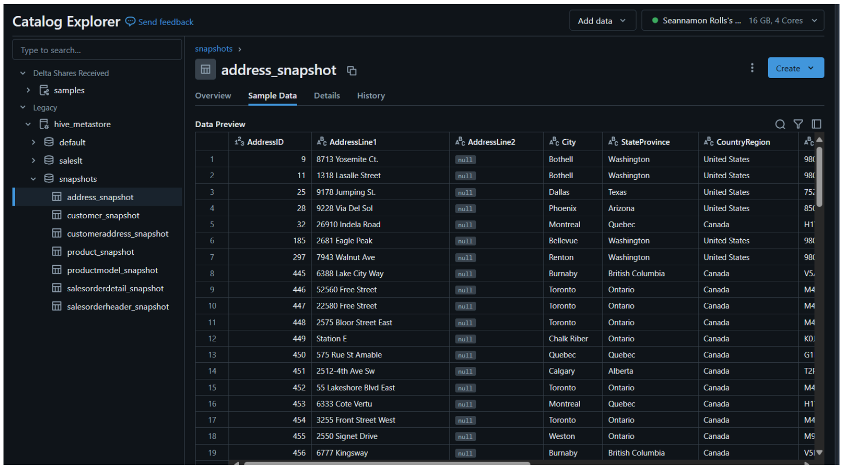
Creating dim_customer.yml

sales.yml

```
version: 2

models:
  - name: dim_sales
    description: This is the fact table for sales
    columns:
      - name: saleOrderID
        description: The surrogate key of the sale order
        tests:
          - unique
          - not_null
      - name: saleOrderDetailID
        description: The surrogate key of the sale order
detail
        tests:
          - unique
          - not_null
      - name: orderQty
        description: The quantity of the order
        tests:
          - not_null
      - name: productID
        description: The surrogate key of the product
        tests:
          - not_null
      - name: unitPrice
        description: The unit price of the product
        tests:
          - not_null
      - name: unitPriceDiscount
        description: The unit price discount of the
product
      - name: lineTotal
        description: The line total of the product
        tests:
          - not_null
      - name: name
        description: The name of the product
        tests:
          - not_null
      - name: productNumber
        description: The product number of the product
        tests:
          - not_null
      - name: color
        description: The color of the product
      - name: standardCost
        description: The standard cost of the product
        tests:
          - not_null
      - name: listPrice
        description: The list price of the product
        tests:
          - not_null
      - name: size
        description: The size of the product
      - name: weight
        description: The weight of the product
      - name: sellStartDate
        description: The date when the product is
available for sale
        tests:
          - not_null
      - name: sellEndDate
        description: The date when the product is no
longer available for sale
      - name: discontinuedDate
        description: The date when the product is
discontinued
      - name: thumbNailPhoto
        description: The thumbnail photo of the product
      - name: thumbnailPhotoFileName
        description: The thumbnail photo file name of the
product
      - name: revisionNumber
        description: The revision number of the sale
order
      - name: orderDate
        description: The order date of the sale order
        tests:
          - not_null
      - name: dueDate
        description: The due date of the sale order
      - name: shipDate
        description: The ship date of the sale order
      - name: status
        description: The status of the sale order
      - name: onlineOrderFlag
        description: The online order flag of the sale
order
      - name: salesOrderNumber
        description: The sales order number of the sale
order
      - name: purchaseOrderNumber
        description: The purchase order number of the
sale order
      - name: accountNumber
        description: The account number of the sale order
      - name: customerID
        description: The surrogate key of the customer
        tests:
          - not_null
      - name: shipToAddressID
        description: The surrogate key of the ship to
address
      - name: billToAddressID
        description: The surrogate key of the bill to
address
      - name: shipMethod
        description: The ship method of the sale order
      - name: creditCardApprovalCode
        description: The credit card approval code of the
sale order
      - name: subTotal
        description: The sub total of the sale order
        tests:
          - not_null
      - name: taxAmt
        description: The tax amount of the sale order
        tests:
          - not_null
      - name: freight
```

```
          description: The freight of the sale order
          tests:
            - not_null
        - name: totalDue
          description: The total due of the sale order
          tests:
            - not_null
        - name: comment
          description: The comment of the sale order
```

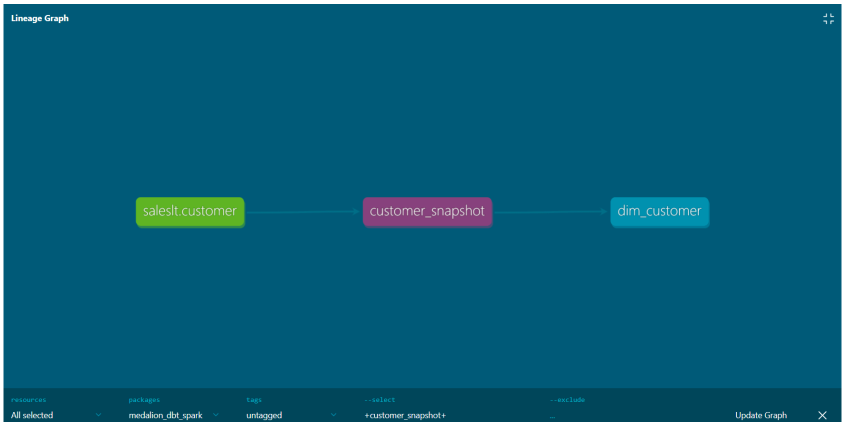
9. Data at silver level result



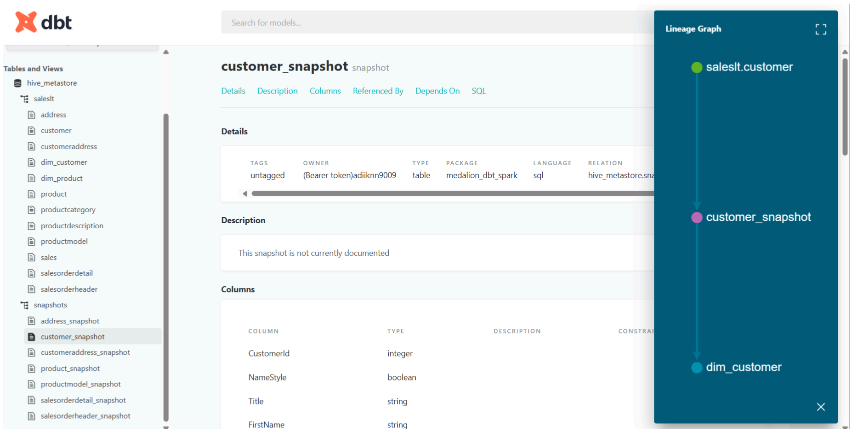
address_snapshot

Data Preview

	AddressID	AddressLine1	AddressLine2	City	StateProvince	CountryRegion	
1	9	8713 Yosemite Ct.	null	Bothell	Washington	United States	980
2	11	1318 Lasalle Street	null	Bothell	Washington	United States	980
3	25	9178 Jumping St.	null	Dallas	Texas	United States	752
4	28	9228 Via Del Sol	null	Phoenix	Arizona	United States	850
5	32	26910 Indela Road	null	Montreal	Quebec	Canada	H1
6	185	2681 Eagle Peak	null	Bellevue	Washington	United States	980
7	297	7943 Walnut Ave	null	Renton	Washington	United States	980
8	445	6388 Lake City Way	null	Burnaby	British Columbia	Canada	V5
9	446	52560 Free Street	null	Toronto	Ontario	Canada	M4
10	447	22580 Free Street	null	Toronto	Ontario	Canada	M4
11	448	2575 Bloor Street East	null	Toronto	Ontario	Canada	M4
12	449	Station E	null	Chalk Riber	Ontario	Canada	K0
13	450	575 Rue St Amable	null	Quebec	Quebec	Canada	G1
14	451	2512-4th Ave Sw	null	Calgary	Alberta	Canada	T2
15	452	55 Lakeshore Blvd East	null	Toronto	Ontario	Canada	M4
16	453	6333 Cote Vertu	null	Montreal	Quebec	Canada	H1
17	454	3255 Front Street West	null	Toronto	Ontario	Canada	M4
18	455	2550 Signet Drive	null	Weston	Ontario	Canada	M9
19	456	6777 Kingsway	null	Burnaby	British Columbia	Canada	V5



resources: All selected
packages: medalion_dbt_spark
tags: untagged
--select: +customer_snapshot+
--exclude: ...
Update Graph



customer_snapshot snapshot

Details Description Columns Referenced By Depends On SQL

Details

TAGS	OWNER	TYPE	PACKAGE	LANGUAGE	RELATION
untagged	(Bearer token)adiiknn9009	table	medalion_dbt_spark	sql	hive_metastore.sna

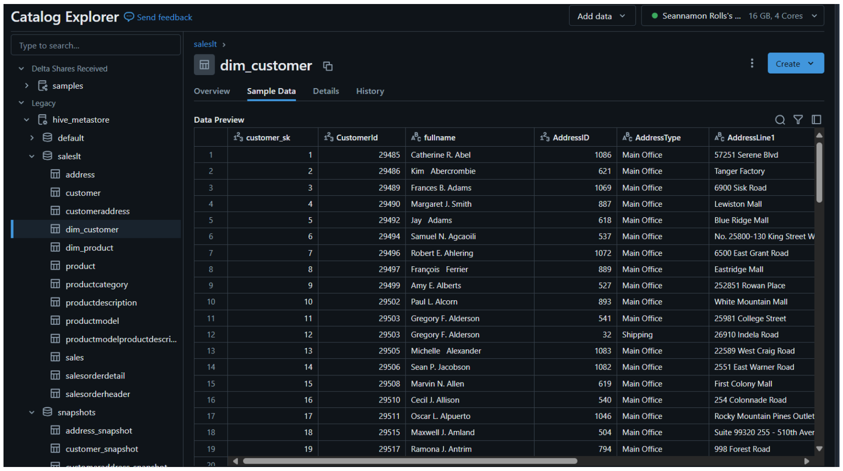
Description

This snapshot is not currently documented

Columns

COLUMN	TYPE	DESCRIPTION	CONSTRAI
CustomerId	integer		
NameStyle	boolean		
Title	string		
FirstName	string		

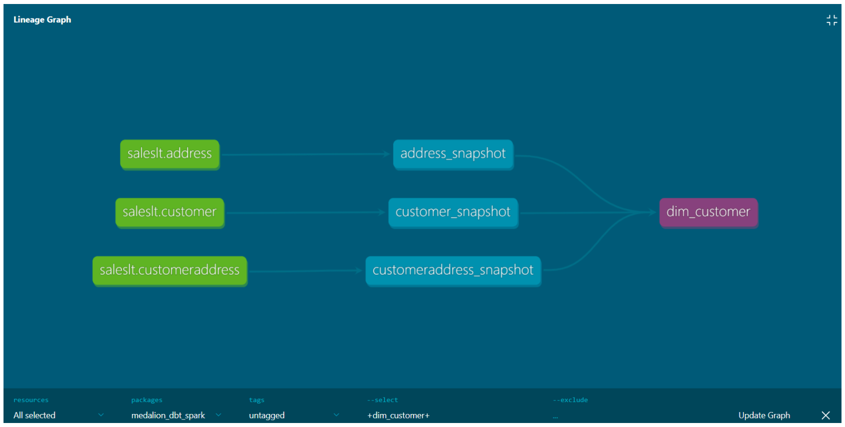
10. Data at gold level result



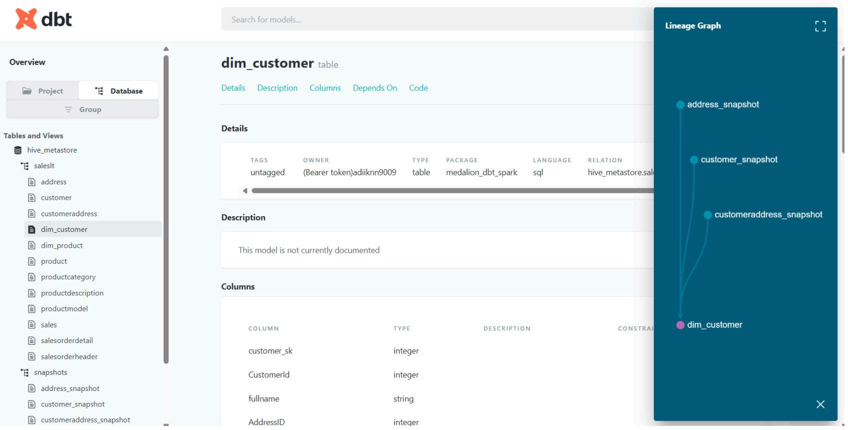
dim_customer

Data Preview

	customer_sk	CustomerId	fullname	AddressID	AddressType	AddressLine1
1	1	29485	Catherine R. Abel	1086	Main Office	57251 Serene Blvd
2	2	29486	Kim Abercrombie	621	Main Office	Tanger Factory
3	3	29489	Frances B. Adams	1069	Main Office	6900 Sisk Road
4	4	29490	Margaret J. Smith	887	Main Office	Lewiston Mall
5	5	29492	Jay Adams	618	Main Office	Blue Ridge Mall
6	6	29494	Samuel N. Agcaoili	537	Main Office	No. 25800-130 King Street W
7	7	29496	Robert E. Ahlering	1072	Main Office	6500 East Grant Road
8	8	29497	François Ferrier	889	Main Office	Eastridge Mall
9	9	29499	Amy E. Alberts	527	Main Office	252851 Rowan Place
10	10	29502	Paul L. Alcorn	893	Main Office	White Mountain Mall
11	11	29503	Gregory F. Alderson	541	Main Office	25981 College Street
12	12	29503	Gregory F. Alderson	32	Shipping	26910 Indela Road
13	13	29505	Michelle Alexander	1083	Main Office	22589 West Craig Road
14	14	29506	Sean P. Jacobson	1082	Main Office	2551 East Warner Road
15	15	29508	Marvin N. Allen	619	Main Office	First Colony Mall
16	16	29510	Cecil J. Allison	540	Main Office	254 Colonnade Road
17	17	29511	Oscar L. Alpuerto	1046	Main Office	Rocky Mountain Pines Outlet
18	18	29515	Maxwell J. Amland	504	Main Office	Suite 99320 255 - 510th Aver
19	19	29517	Ramona J. Antrim	794	Main Office	998 Forest Road
20						



resources: All selected
packages: medalion_dbt_spark
tags: untagged
--select: +dim_customer+
--exclude: ...



dim_customer table

Details

TAGS	OWNER	TYPE	PACKAGE	LANGUAGE	RELATION
untagged	(Bearer token)adiiknn9009	table	medalion_dbt_spark	sql	hive_metastore.sal

Description

This model is not currently documented

Columns

COLUMN	TYPE	DESCRIPTION	CONSTRAI
customer_sk	integer		
CustomerId	integer		
fullname	string		
AddressID	integer		