Chinook OLTP → Snowflake Data Warehouse Integration using Azure Data Factory

1. Objective

The goal of this project is to replicate the *Chinook OLTP to Data Warehouse* data integration workflow using **Azure Data Factory (ADF)**, **Azure SQL Database**, **Azure Blob Storage**, and **Snowflake**.

The deliverable demonstrates an end-to-end ETL process that extracts data from the Chinook transactional database hosted in Azure SQL, stages it in Azure Blob Storage as Parquet files, and then loads it into Snowflake's `STAGE` and `DW` schemas using ADF pipelines and data flows.

2. Architecture Overview

The architecture follows a layered ELT approach:

1. **Extract Layer (ADF Pipeline 1)**
    - Source: Azure SQL Database (Chinook OLTP)
    - Destination: Azure Blob Storage (Parquet format)
2. **Stage Layer (ADF Pipeline 2)**
    - Reads Parquet data from Blob Storage
    - Loads into Snowflake `STAGE` schema tables
3. **Transformation Layer (ADF Data Flow)**
    - Performs transformation logic (hashing, deduplication, merge updates)
    - Loads transformed data into Snowflake `DW` schema tables
4. **Data Warehouse Layer**
    - Snowflake hosts final `DATE_DIM`, `TIME_DIM`, `CUSTOMER_DIM`, `ARTIST_DIM`, and `SALES_FACT` tables.
    - These form a star schema suitable for business analytics.

3. Tools and Technologies Used

Component	Purpose
Azure SQL Database	Stores source Chinook OLTP tables (Customer, Invoice, Artist, Album)
Azure Data Factory (ADF)	Orchestrates extraction, staging, and loading via pipelines and data flows
Azure Blob Storage	Intermediate storage for Parquet files
Azure Key Vault	Secures Snowflake and SQL credentials
Snowflake	Target cloud data warehouse hosting `STAGE` and `DW` schemas
DBeaver / Snowflake UI	Used for running SQL DDL, validation, and testing

4. Azure Setup

Screenshot: `Setup.png`

Created the following resources:

- SQL Server: `damg7370fall2025.database.windows.net`
- SQL Database: `DAMG7370FALL2025`
- Storage Account: `stgchinookdamg`
- Azure Data Factory: `adfchinookdamg`
- Azure Key Vault: `AzureKeyVault1`

Networking setup:

- Added client IP address to firewall
- Enabled “Allow Azure services to access server”
- Verified connection via SSMS and ADF

5. Datasets

Screenshots: `Dataset1.png`, `Dataset2.png`

Dataset Name	Type	Linked Service	Purpose
`sqlserverdb_chinook`	Azure SQL Database	`Sql_db_Chinook`	Source dataset
`Parquet_ds`	Parquet (Blob)	`Storage_Chinook`	Staging Parquet output
`SnowChinook_Ds`	Snowflake	`snow_chinook`	Target dataset for loading into STAGE schema

6. Pipeline 1 — Extract SQL DB to Parquet

Screenshot: `1stpipeline.png`

Pipeline Name: `extract_SQLDB_PL`

- Uses a **ForEach** activity looping through the array: `["Customer", "Artist", "Album", "Invoice"]`
- Inside loop: Copy Activity (`sql_2_parquet`)
  - Source: Azure SQL Database
  - Sink: Azure Blob Storage (Parquet)
- Managed identity access configured for Blob Storage.
- All activities succeeded with matching row counts.

8. Pipeline 2 — Parquet to Snowflake Stage

Screenshot: `2nd pipeline.png`

Pipeline Name: `Parquet_2_SnowStage_PL`

- Reads Parquet files from Blob container `/stage_data/`
- Writes into Snowflake `STAGE` tables:
  - `STAGE.CUSTOMER`
  - `STAGE.ARTIST`
  - `STAGE.ALBUM`
  - `STAGE.INVOICE`
- Connected using SAS-based Snowflake Linked Service for stage loading.

9. Data Flow — Load Customer Dimension

Screenshot: `Dataflow.png`

Data Flow Name: `DF_Load_Customer_DIM`

Steps:

1. **Source:** `STAGE.CUSTOMER`
2. **Derived Column:** Generate `CUSTOMER_HASH` using SHA-256 for change detection.
3. **Join:** Compare incoming vs existing customers.
4. **Sink:** Upsert into `DW.CUSTOMER_DIM`
   - Merge logic based on hash value.
   - Insert new and update changed records.

10. Snowflake Schemas and Tables

Screenshot: `Dimtables.png`

Executed SQL scripts in Snowflake:

1. `create_stage_schema.sql`
2. `create_dw_schema.sql`
3. `load_date_dim.sql`
4. `load_time_dim.sql`
5. `merge_artist_dim.sql`
6. `load_sales_fact.sql`
7. `validation_counts.sql`

Final Star Schema Tables:

- **Dimensions:**
  `DATE_DIM`, `TIME_DIM`, `CUSTOMER_DIM`, `ARTIST_DIM`
- **Fact Table:**
  `SALES_FACT` (references DATE_DIM_KEY and CUSTOMER_KEY)

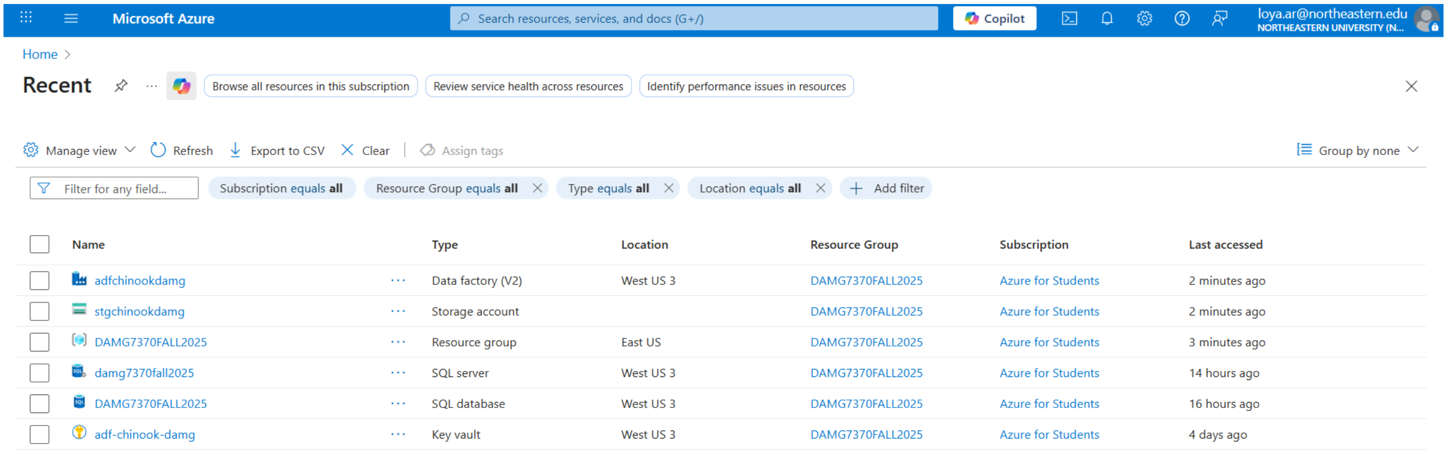
11. SQL Script Summary

Script	Purpose
`create_stage_schema.sql`	Defines staging layer tables
`create_dw_schema.sql`	Creates DW layer (DIM & FACT)
`load_date_dim.sql`	Populates `DATE_DIM`
`load_time_dim.sql`	Populates `TIME_DIM`

`merge_artist_dim.sql`	Incremental load for `ARTIST_DIM`
`load_sales_fact.sql`	Inserts transactional data into `SALES_FACT`
`validation_counts.sql`	Validates record counts across all layers

All Screenshots

SETUP

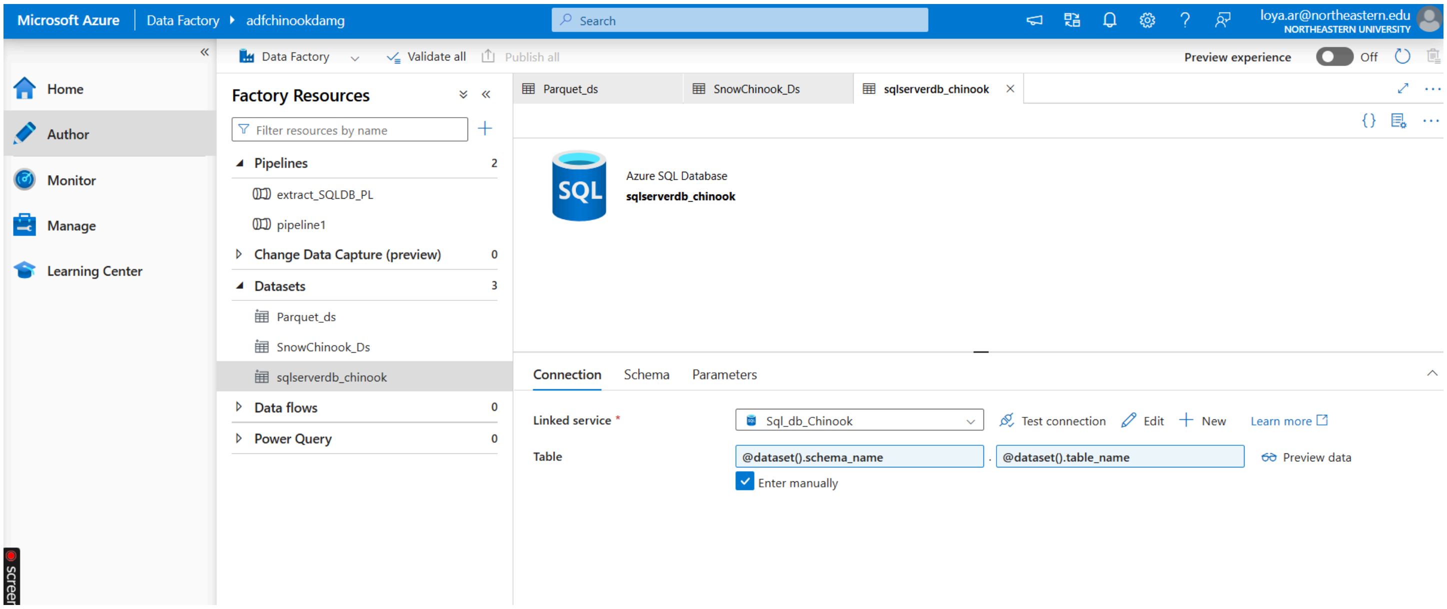

Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
loya.ar@northeastern.edu
NORTHEASTERN UNIVERSITY (N...
Home
Browse all resources in this subscription
Review service health across resources
Identify performance issues in resources
Manage view
Refresh
Export to CSV
Clear
Assign tags
Group by none
Filter for any field...
Add filter

Recent

Subscription equals **all** · Resource Group equals **all** · Type equals **all** · Location equals **all**

Name	Type	Location	Resource Group	Subscription	Last accessed
adfchinookdamg	Data factory (V2)	West US 3	DAMG7370FALL2025	Azure for Students	2 minutes ago
stgchinookdamg	Storage account		DAMG7370FALL2025	Azure for Students	2 minutes ago
DAMG7370FALL2025	Resource group	East US	DAMG7370FALL2025	Azure for Students	3 minutes ago
damg7370fall2025	SQL server	West US 3	DAMG7370FALL2025	Azure for Students	14 hours ago
DAMG7370FALL2025	SQL database	West US 3	DAMG7370FALL2025	Azure for Students	16 hours ago
adf-chinook-damg	Key vault	West US 3	DAMG7370FALL2025	Azure for Students	4 days ago

DATA SET 1



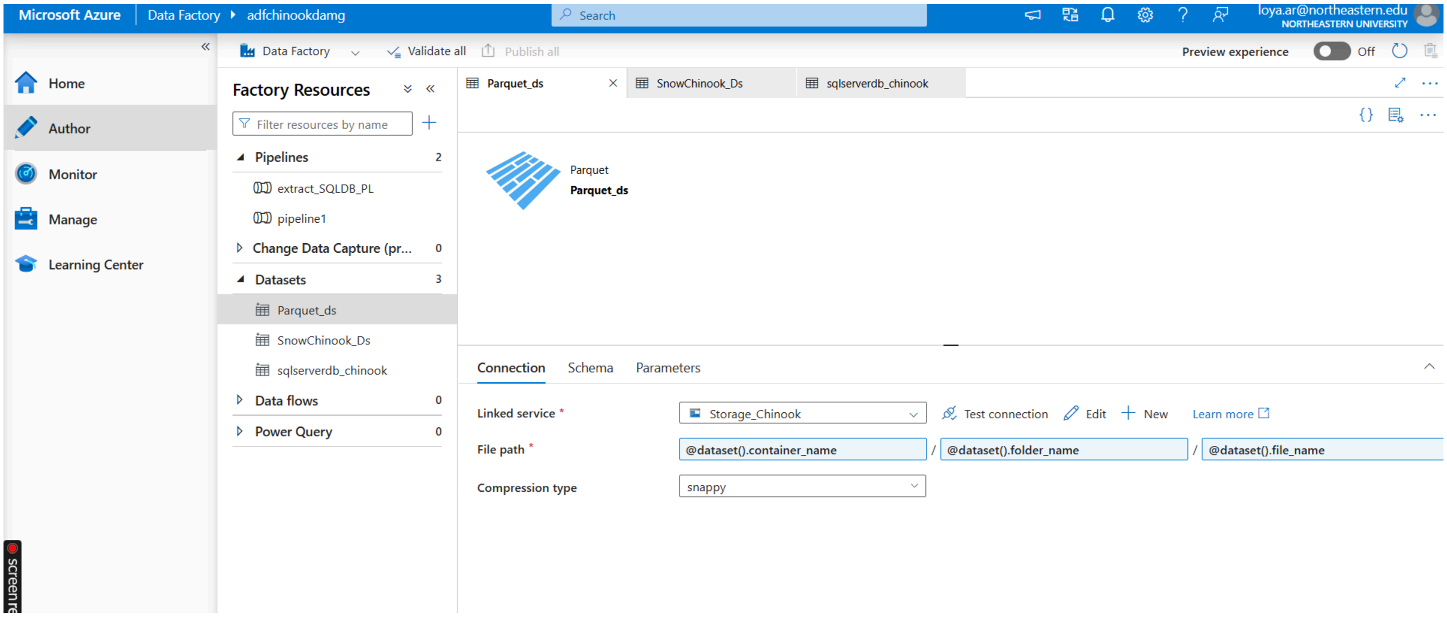
Azure SQL Database
sqlserverdb_chinook

Linked service * : Sql_db_Chinook

Table: @dataset().schema_name . @dataset().table_name

☑ Enter manually

DATASET 2

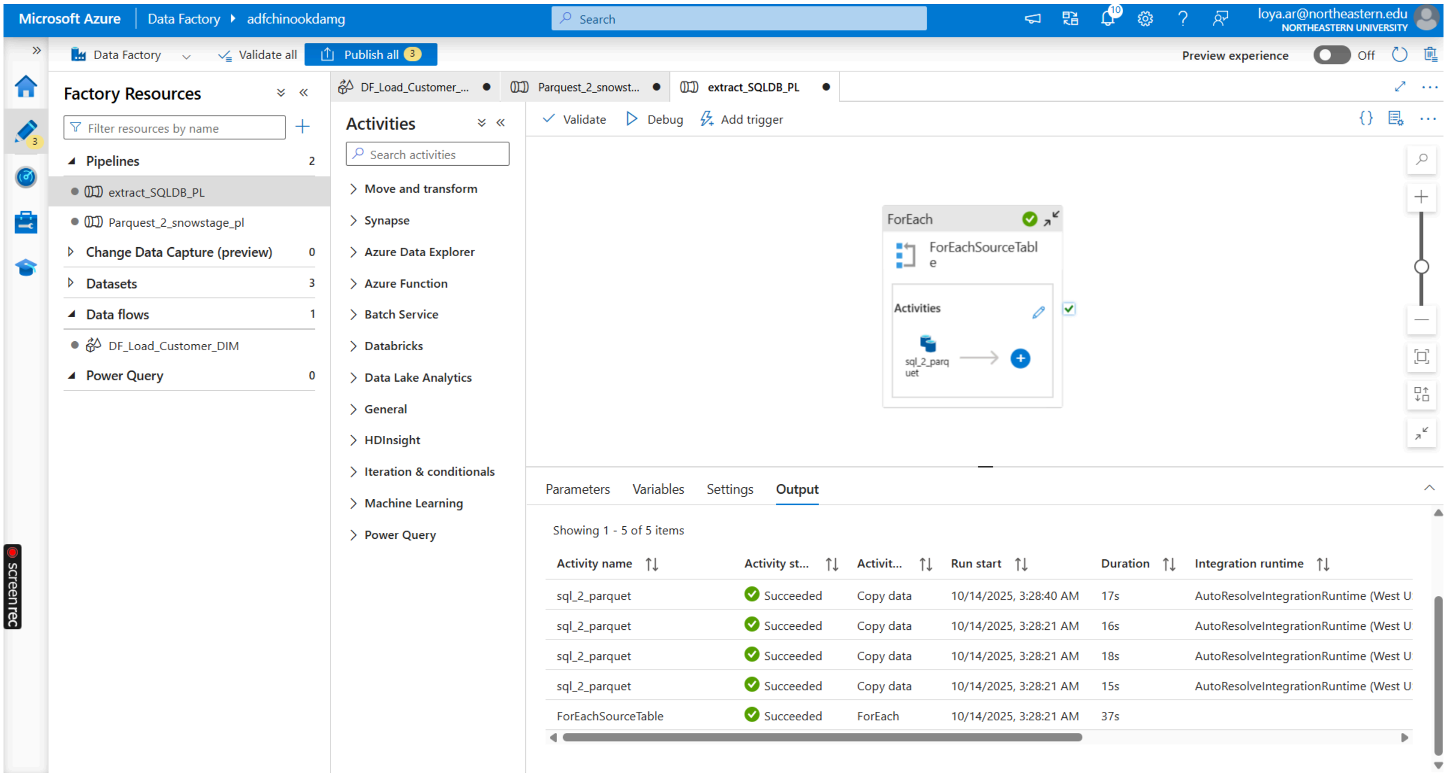


Linked service * Storage_Chinook

File path * @dataset().container_name / @dataset().folder_name / @dataset().file_name

Compression type snappy

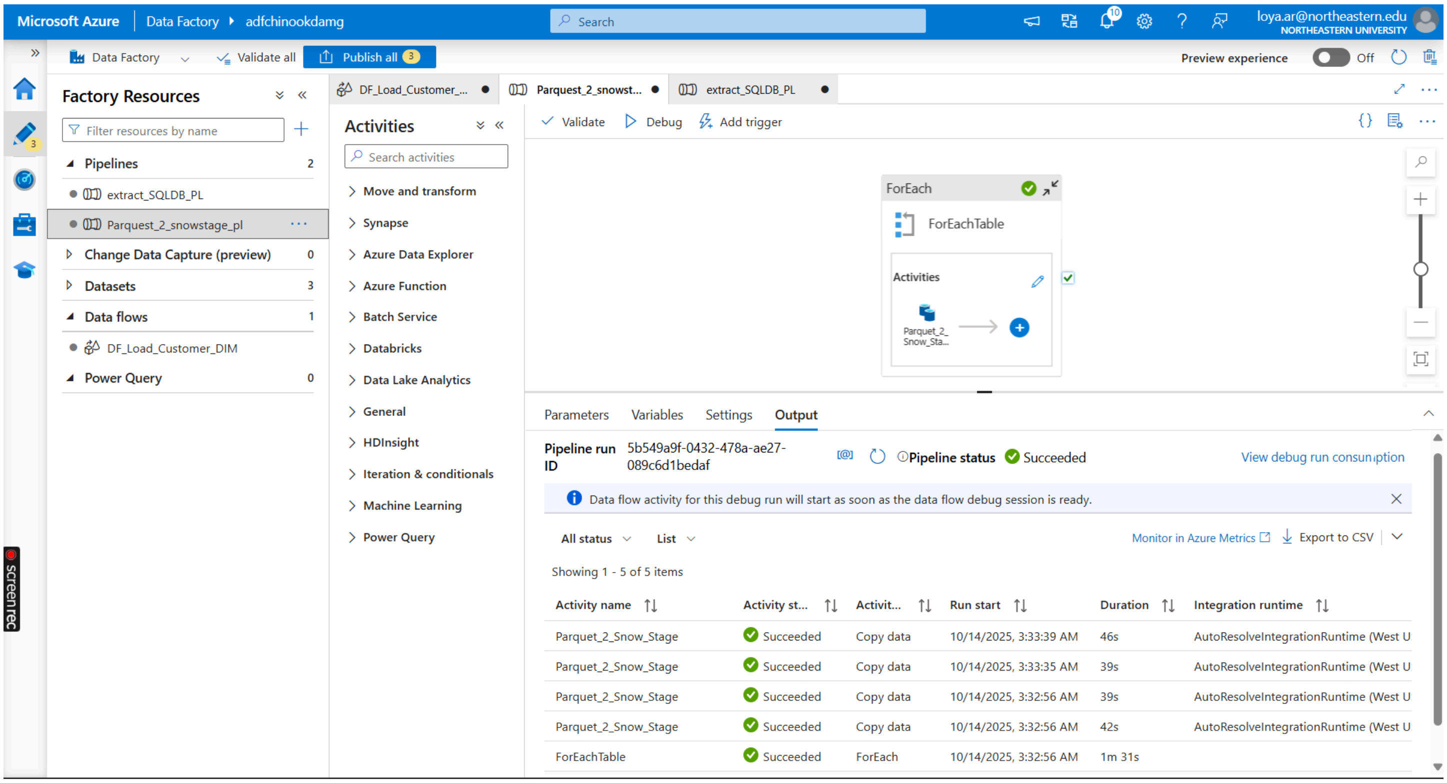
PIPELINE- 1



Showing 1 - 5 of 5 items

Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime
sql_2_parquet	Succeeded	Copy data	10/14/2025, 3:28:40 AM	17s	AutoResolveIntegrationRuntime (West U
sql_2_parquet	Succeeded	Copy data	10/14/2025, 3:28:21 AM	16s	AutoResolveIntegrationRuntime (West U
sql_2_parquet	Succeeded	Copy data	10/14/2025, 3:28:21 AM	18s	AutoResolveIntegrationRuntime (West U
sql_2_parquet	Succeeded	Copy data	10/14/2025, 3:28:21 AM	15s	AutoResolveIntegrationRuntime (West U
ForEachSourceTable	Succeeded	ForEach	10/14/2025, 3:28:21 AM	37s	

PIPELINE-2



Pipeline run ID 5b549a9f-0432-478a-ae27-089c6d1bedaf

Pipeline status ✓ Succeeded

View debug run consumption

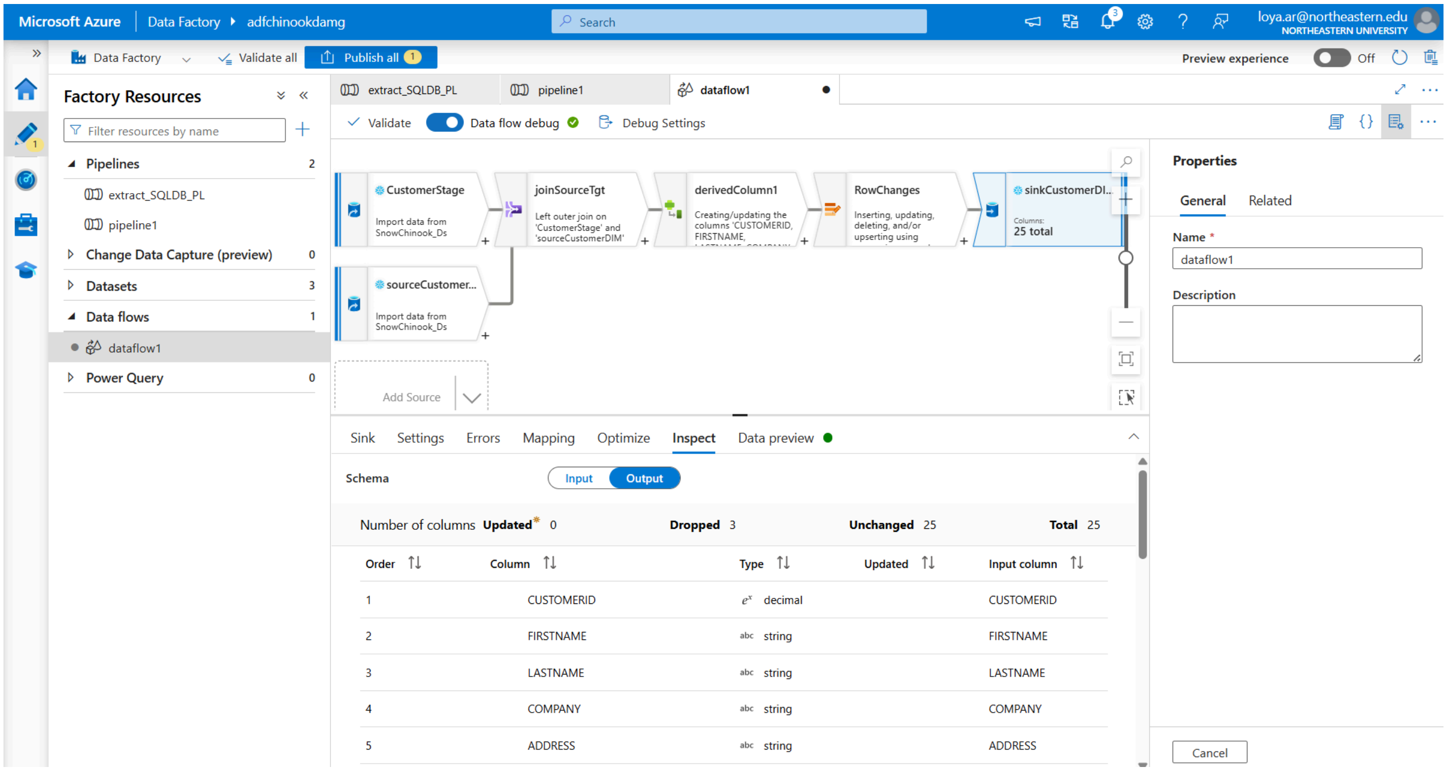
Data flow activity for this debug run will start as soon as the data flow debug session is ready.

All status | List | Monitor in Azure Metrics | Export to CSV

Showing 1 - 5 of 5 items

Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime
Parquet_2_Snow_Stage	✓ Succeeded	Copy data	10/14/2025, 3:33:39 AM	46s	AutoResolveIntegrationRuntime (West U
Parquet_2_Snow_Stage	✓ Succeeded	Copy data	10/14/2025, 3:33:35 AM	39s	AutoResolveIntegrationRuntime (West U
Parquet_2_Snow_Stage	✓ Succeeded	Copy data	10/14/2025, 3:32:56 AM	39s	AutoResolveIntegrationRuntime (West U
Parquet_2_Snow_Stage	✓ Succeeded	Copy data	10/14/2025, 3:32:56 AM	42s	AutoResolveIntegrationRuntime (West U
ForEachTable	✓ Succeeded	ForEach	10/14/2025, 3:32:56 AM	1m 31s	

DATAFLOW



Name * dataflow1

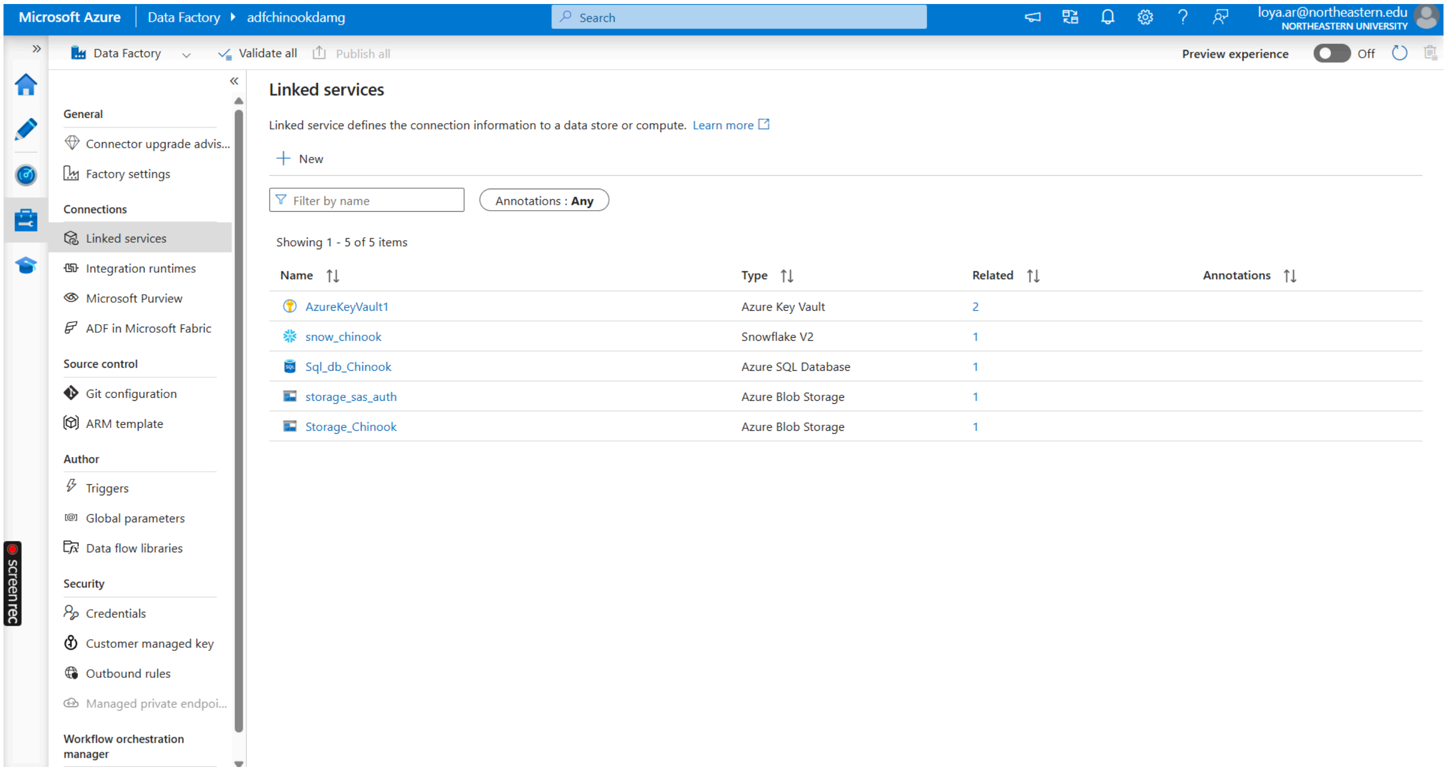
Description

Schema: Input | Output

Number of columns Updated* 0 Dropped 3 Unchanged 25 Total 25

Order	Column	Type	Updated	Input column
1	CUSTOMERID	decimal		CUSTOMERID
2	FIRSTNAME	string		FIRSTNAME
3	LASTNAME	string		LASTNAME
4	COMPANY	string		COMPANY
5	ADDRESS	string		ADDRESS

LINKED SERVICES



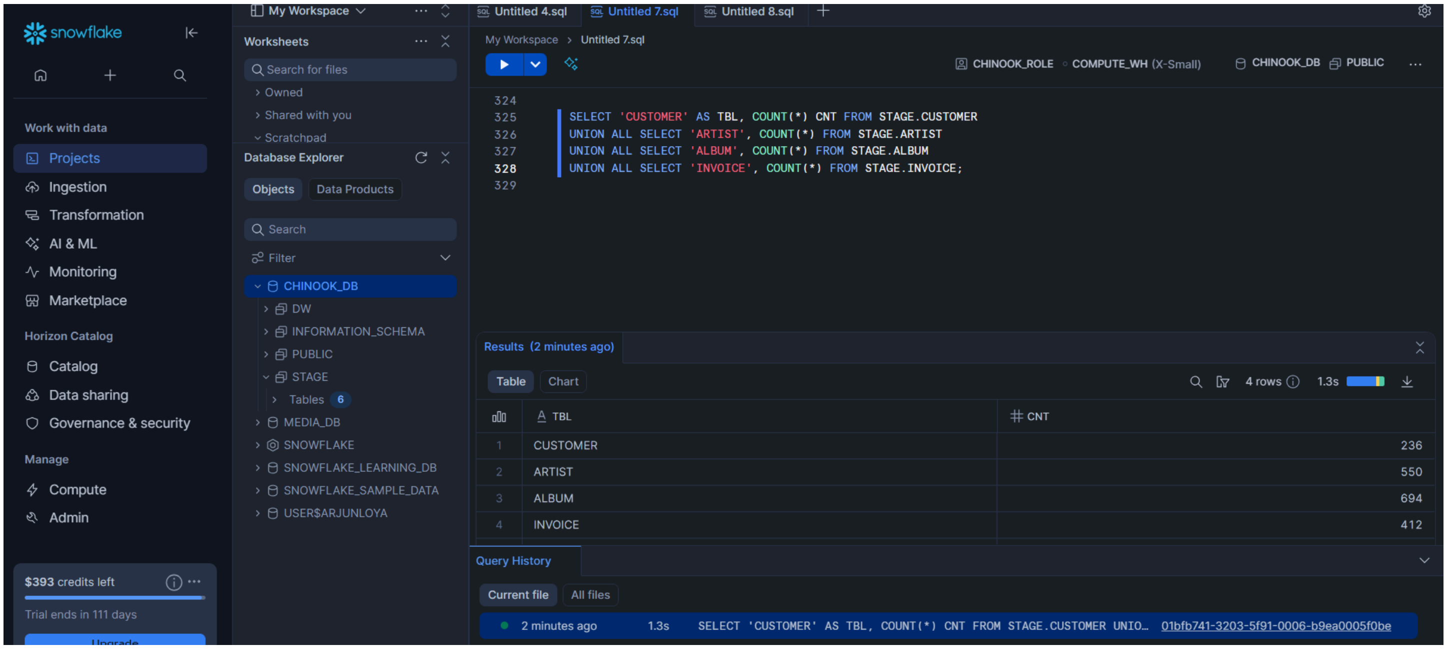
Linked services

Linked service defines the connection information to a data store or compute. Learn more

Showing 1 - 5 of 5 items

Name	Type	Related	Annotations
AzureKeyVault1	Azure Key Vault	2	
snow_chinook	Snowflake V2	1	
Sql_db_Chinook	Azure SQL Database	1	
storage_sas_auth	Azure Blob Storage	1	
Storage_Chinook	Azure Blob Storage	1	

VALIDATION

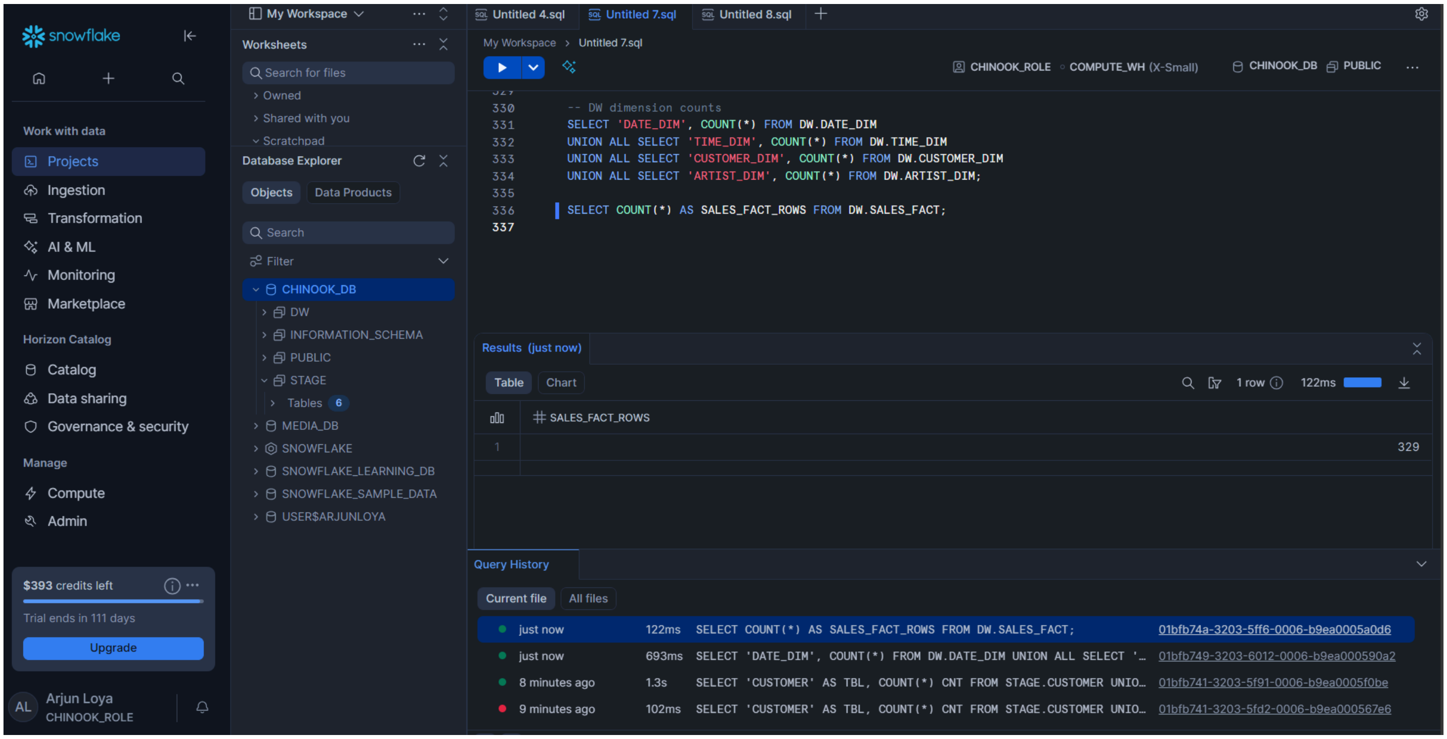


```
SELECT 'CUSTOMER' AS TBL, COUNT(*) CNT FROM STAGE.CUSTOMER
UNION ALL SELECT 'ARTIST', COUNT(*) FROM STAGE.ARTIST
UNION ALL SELECT 'ALBUM', COUNT(*) FROM STAGE.ALBUM
UNION ALL SELECT 'INVOICE', COUNT(*) FROM STAGE.INVOICE;
```

	TBL	CNT
1	CUSTOMER	236
2	ARTIST	550
3	ALBUM	694
4	INVOICE	412

```
● 2 minutes ago    1.3s    SELECT 'CUSTOMER' AS TBL, COUNT(*) CNT FROM STAGE.CUSTOMER UNIO…  01bfb741-3203-5f91-0006-b9ea0005f0be
```

DIMTABLES



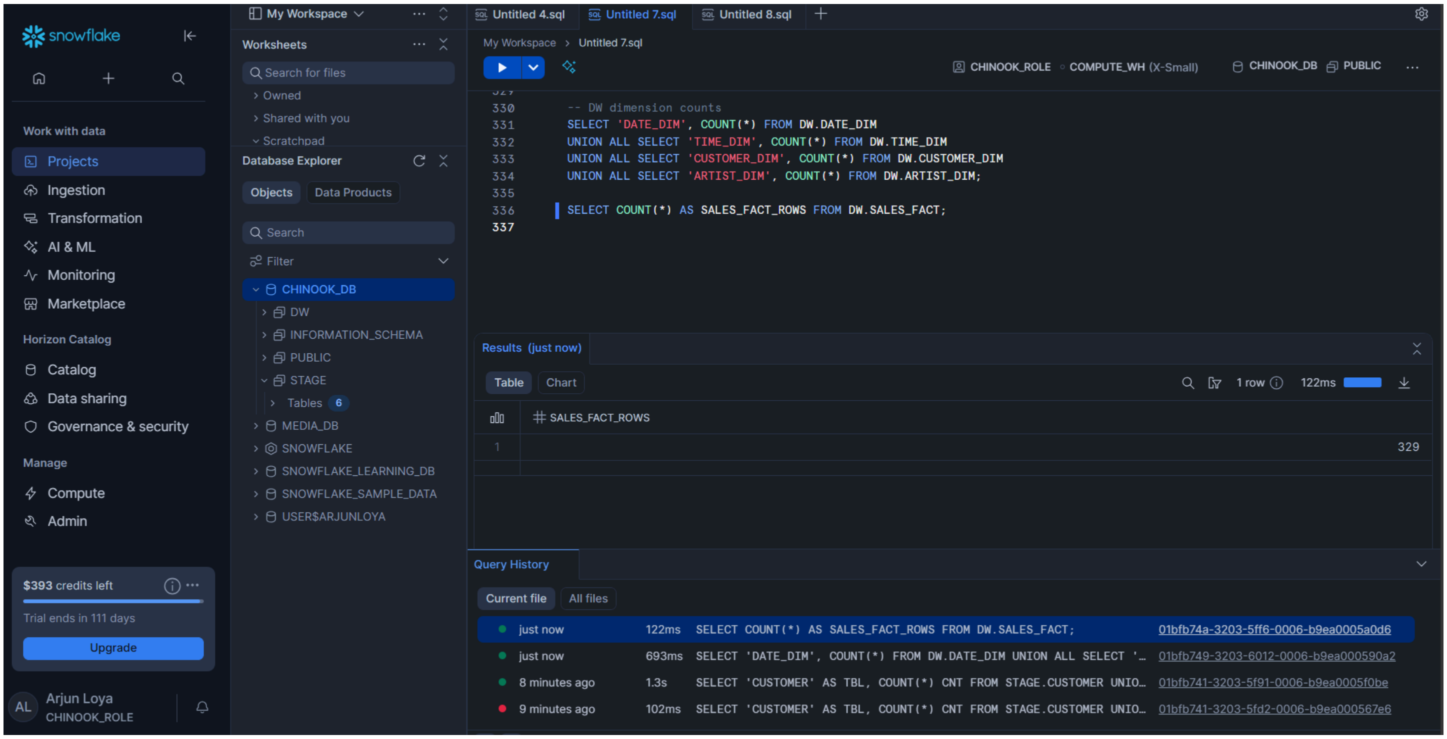
```sql
-- DW dimension counts
SELECT 'DATE_DIM', COUNT(*) FROM DW.DATE_DIM
UNION ALL SELECT 'TIME_DIM', COUNT(*) FROM DW.TIME_DIM
UNION ALL SELECT 'CUSTOMER_DIM', COUNT(*) FROM DW.CUSTOMER_DIM
UNION ALL SELECT 'ARTIST_DIM', COUNT(*) FROM DW.ARTIST_DIM;

SELECT COUNT(*) AS SALES_FACT_ROWS FROM DW.SALES_FACT;
```

	SALES_FACT_ROWS
1	329

just now	122ms	SELECT COUNT(*) AS SALES_FACT_ROWS FROM DW.SALES_FACT;	01bfb74a-3203-5ff6-0006-b9ea0005a0d6
just now	693ms	SELECT 'DATE_DIM', COUNT(*) FROM DW.DATE_DIM UNION ALL SELECT '…	01bfb749-3203-6012-0006-b9ea000590a2
8 minutes ago	1.3s	SELECT 'CUSTOMER' AS TBL, COUNT(*) CNT FROM STAGE.CUSTOMER UNIO…	01bfb741-3203-5f91-0006-b9ea0005f0be
9 minutes ago	102ms	SELECT 'CUSTOMER' AS TBL, COUNT(*) CNT FROM STAGE.CUSTOMER UNIO…	01bfb741-3203-5fd2-0006-b9ea000567e6

FACT_TABLE_VALIDATION



```sql
-- DW dimension counts
SELECT 'DATE_DIM', COUNT(*) FROM DW.DATE_DIM
UNION ALL SELECT 'TIME_DIM', COUNT(*) FROM DW.TIME_DIM
UNION ALL SELECT 'CUSTOMER_DIM', COUNT(*) FROM DW.CUSTOMER_DIM
UNION ALL SELECT 'ARTIST_DIM', COUNT(*) FROM DW.ARTIST_DIM;

SELECT COUNT(*) AS SALES_FACT_ROWS FROM DW.SALES_FACT;
```

	SALES_FACT_ROWS
1	329

just now	122ms	SELECT COUNT(*) AS SALES_FACT_ROWS FROM DW.SALES_FACT;	01bfb74a-3203-5ff6-0006-b9ea0005a0d6
just now	693ms	SELECT 'DATE_DIM', COUNT(*) FROM DW.DATE_DIM UNION ALL SELECT '…	01bfb749-3203-6012-0006-b9ea000590a2
8 minutes ago	1.3s	SELECT 'CUSTOMER' AS TBL, COUNT(*) CNT FROM STAGE.CUSTOMER UNIO…	01bfb741-3203-5f91-0006-b9ea0005f0be
9 minutes ago	102ms	SELECT 'CUSTOMER' AS TBL, COUNT(*) CNT FROM STAGE.CUSTOMER UNIO…	01bfb741-3203-5fd2-0006-b9ea000567e6

THANKYOU