

# End-To-End Data Engineering Project using Microsoft Azure and Azure Databricks

## Business Problem

### Context:

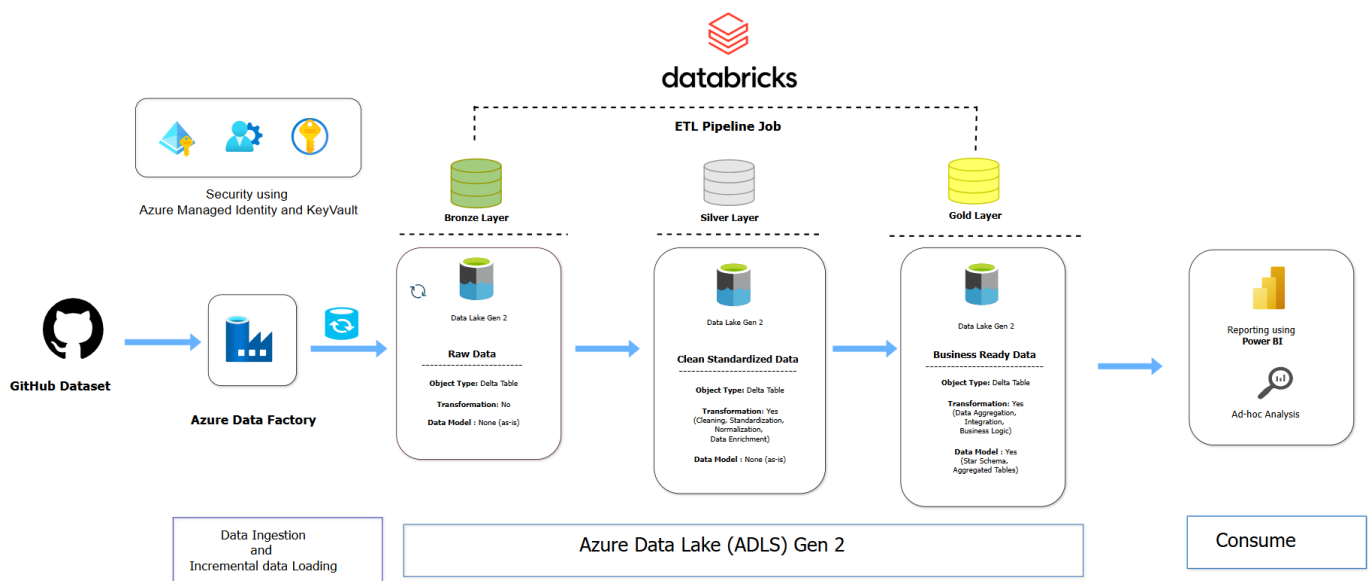
A retail company (or a similar enterprise in this case study) operates in multiple regions with diverse product lines and a network of resellers. Sales are driven by a distributed salesforce, and the company sets quarterly/annual sales targets at different levels (by product, region, or salesperson). However, the company faces challenges in tracking performance, identifying underperforming regions/resellers, and aligning sales strategy with business targets.

### Key Business Problems:

- Lack of unified view of sales across regions, products, and resellers.
- Difficulty in tracking sales performance vs. targets.
- Inefficient allocation of resources to underperforming areas.
- Limited forecasting and strategy alignment due to poor data insights.


### Proposed Data-Driven Solution:








- Ingest raw CSV datasets into **Azure Data Lake Gen2** using **Azure Data Factory**.
- Store raw, untransformed data in the **Bronze Layer** (Delta Tables) in Data Lake.
- Apply cleaning, standardization, and enrichment in the **Silver Layer**.
- Aggregate and apply business logic in the **Gold Layer** (Star Schema).
- Secure the pipeline with **Azure Managed Identity and Key Vault**.
- Deliver insights via **Power BI dashboard**.



# GitHub Dataset

retail-analytics-pipeline-azure-databricks-project / data / [Add file](#) [...](#)


 **developersview** updated files e92084c · 2 weeks ago [History](#)

Name	Last commit message	Last commit date
..		
 Product.csv	updated files	2 weeks ago
 Region.csv	updated data	2 weeks ago
 Reseller.csv	updated files	2 weeks ago
 Sales.csv	updated files	2 weeks ago
 Salesperson.csv	updated data	2 weeks ago
 SalespersonRegion.csv	updated data	2 weeks ago
 Targets.csv	updated data	2 weeks ago

## Resource Group

Resources used:

1. Azure Databricks
2. Azure Data Factory (ADF)
3. Azure Data Lake Storage (ADLS) Gen 2
4. Azure Databricks Access Connector

 **RG-DEProject** [How do I troubleshoot issues with this resource group?](#) [How do I monitor this resource group?](#) [+1](#) [X](#)






Search [Create](#) [Manage view](#) [Delete resource group](#) [Refresh](#) [Export to CSV](#) [Open query](#) [Assign tags](#) [...](#)

**Overview** [Essentials](#) [JSON View](#)

**Resources** Recommendations (1)

Filter for any field... [Type equals all](#) [Location equals all](#) [Add filter](#)

Showing 1 to 5 of 5 records. ☐ Show hidden types [No grouping](#) [List view](#)

<input type="checkbox"/> Name ↑↓	Type ↑↓	Location ↑↓	
<input type="checkbox"/>  adb-retail-project-prd	Azure Databricks Service	West US 2	...
<input type="checkbox"/>  adf-retail-project-prd	Data factory (V2)	West US 2	...
<input type="checkbox"/>  azureprojectdatalakegen2	Storage account	West US	...
<input type="checkbox"/>  retail-project-access-connector	Access Connector for Azure Data...	West US 2	...
<input type="checkbox"/>  syn-retail-prd	Synapse workspace	UK South	...

[< Previous](#) Page [1](#) of 1 [Next >](#) [Give feedback](#)

# Azure Data Lake Storage (ADLS) Gen 2

Home > azureprojectdatalakegen2

azureprojectdatalakegen2 | Containers ☆ ...

Storage account

Search

+ Add container ↑ Upload ↻ Refresh | 🗑 Delete 🔒 Change access level ↺ Restore containers ▾ 🛠 Edit columns

Search containers by prefix

Only show active containers

Showing all 5 items

<input type="checkbox"/>	Name	Last modified	Anonymous access level	Lease state
<input type="checkbox"/>	\$logs	8/9/2025, 7:00:24 PM	Private	Available
<input type="checkbox"/>	bronze	8/9/2025, 7:04:06 PM	Private	Available
<input type="checkbox"/>	gold	8/9/2025, 7:04:28 PM	Private	Available
<input type="checkbox"/>	lookup	8/22/2025, 5:04:55 PM	Private	Available
<input type="checkbox"/>	silver	8/9/2025, 7:04:20 PM	Private	Available

Overview

Activity log

Tags

Diagnose and solve problems

Access Control (IAM)

Data migration

Events

Storage browser

Partner solutions

Resource visualizer

Data storage

Containers

File shares

Queues

Tables

## Azure Data Factory Pipeline – Data Ingestion + Incremental Data Loading

1. Use Lookup Activity to check the Last load value
2. Use another Lookup activity to check Latest Load value
3. Use Copy Activity nested in a ForEach Activity to copy data from GitHub or Any other source
4. Use Copy Activity to update the Last load value with Latest Load

All pipeline runs > ✔ retail-pipeline-ingestdata-prd - Activity runs

🔄 Rerun ▾ ⏸ Cancel ▾ ↻ Refresh 🛠 Update pipeline List Gantt

Activity runs

All status ▾ List ▾ Monitor in Azure Metrics 📊 View run detail 📄 Export to CSV ▾

Showing 1 - 11 items

Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime	User prop...	Activity run ID
LastLoad	✔ Succeeded	Lookup	8/22/2025, 6:07:39 PM	16s	AutoResolveIntegrationRuntime (West US)		0b2d2e88-abd6-4dd8-927e-7cf6070032c6
LatestLoad	✔ Succeeded	Lookup	8/22/2025, 6:07:39 PM	11s	AutoResolveIntegrationRuntime (West US)		cfcd00c9-e5da-40f1-9cfb-3d8cd7a335db
ForEach1	✔ Succeeded	ForEach	8/22/2025, 6:07:57 PM	30s			9b0c7ec1-503a-48cd-abab-2d97d3bd4427
Copy data from github	✔ Succeeded	Copy data	8/22/2025, 6:07:58 PM	25s	AutoResolveIntegrationRuntime (West US)		c721fa6e-2835-4f67-b1e4-df346d869145
Copy data from github	✔ Succeeded	Copy data	8/22/2025, 6:07:58 PM	22s	AutoResolveIntegrationRuntime (West US)		464fde93-77a8-40aa-b821-39a73818351d
Copy data from github	✔ Succeeded	Copy data	8/22/2025, 6:07:58 PM	19s	AutoResolveIntegrationRuntime (West US)		4c6bdcf8-2a6c-429a-a068-dc11ce8129b1
Copy data from github	✔ Succeeded	Copy data	8/22/2025, 6:07:58 PM	18s	AutoResolveIntegrationRuntime (West US)		8b968160-7396-4942-b41e-0d0641ce9b8d

# Azure Databricks Access Connector

Azure Databricks Access Connector is required to access Azure Data Lake from Azure Databricks, without the access connector, Azure Databricks won't be able read / write data from / to Azure Databricks

Home > **retail-project-access-connector** Access Connector for Azure Databricks

Search Refresh Delete

**Overview**

- Activity log
- Access control (IAM)
- Tags
- Resource visualizer
- Settings

**Essentials**

Resource group (move) : [RG-DEProject](#) State :  
 Location : West US 2 Resource ID :  
 Subscription (move) : [Azure subscription 1](#)  
 Subscription ID :  
 Tags (edit) : application : databricks cost center : None env : PRD owner : Pranoy Chakraborty

Home > [azureprojectdatalakegen2](#)

**azureprojectdatalakegen2** | Access Control (IAM) Storage account

Search Add Download role assignments Edit columns Refresh Delete Feedback

Number of role assignments for this subscription 16 4000

Search by name or email Type: All Role: All Scope: All scopes Group by: Role

All (6) Job function roles (4) Privileged administrator roles (2)

Name	Type	Role	Scope	Condition
> Owner (2)				
Storage Blob Data Contributor (4)				
<b>Pranoy Chakraborty</b>	User	Storage Blob Data Contr...	This resource	Add
<b>retail-project-access-connector</b>	Managed identity	Storage Blob Data Contr...	This resource	Add

## Azure Databricks Credentials

Microsoft Azure **databricks** Search data, notebooks, recents, and more... CTRL + P

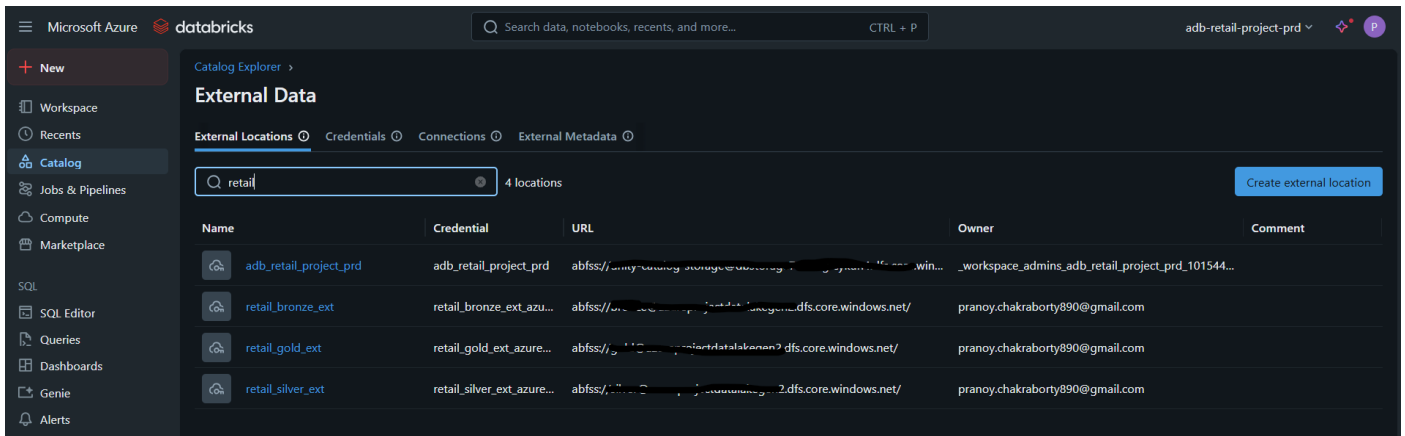
**External Data**

External Locations Credentials Connections External Metadata

Search retail 4 credentials

Name	Purpose	Credential Type	Properties	Owner
<b>adb_retail_project_prd</b>	STORAGE	Managed Identity	Connector Id: [redacted] User Assigned Managed Identity Id: [redacted]	_workspace_admins_adb_retail_...
<b>retail_bronze_ext_azurem...</b>	STORAGE	Managed Identity	Connector Id: [redacted] User Assigned Managed Identity Id: [redacted]	pranoy.chakraborty890@gmail...
<b>retail_gold_ext_azureman...</b>	STORAGE	Managed Identity	Connector Id: [redacted] User Assigned Managed Identity Id: [redacted]	pranoy.chakraborty890@gmail...
<b>retail_silver_ext_azuremar...</b>	STORAGE	Managed Identity	Connector Id: [redacted] User Assigned Managed Identity Id: [redacted]	pranoy.chakraborty890@gmail...

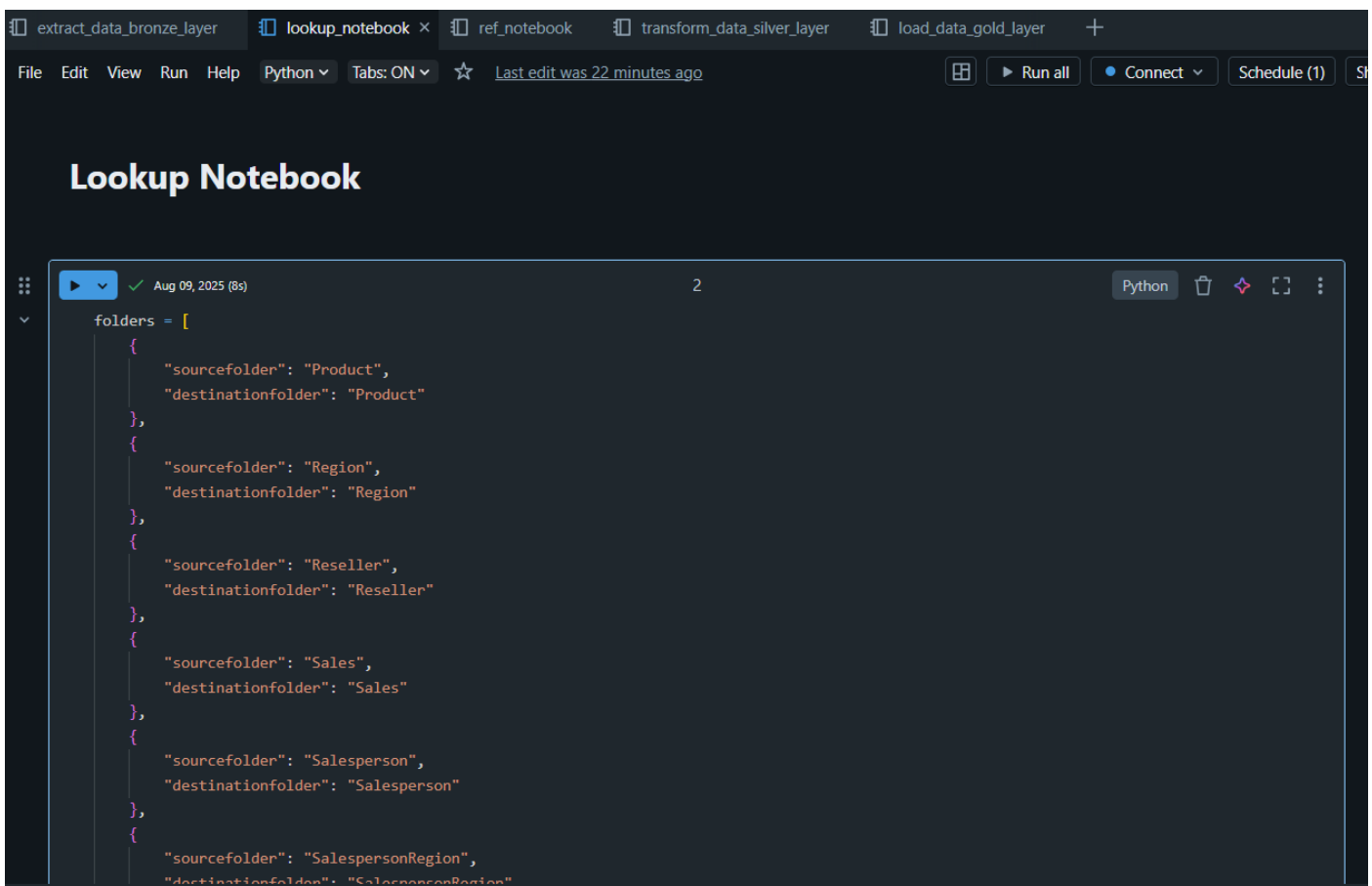
## Azure Databricks External Location



## ETL Pipeline in Azure Databricks

### Lookup Notebook

The Lookup Notebook is used to set a list of files and folders. The output is used in Data Transformation from Bronze to Silver Layer.



## Extract Data from Bronze Layer

Extract and Load data from Bronze layer to Silver layer:

The screenshot shows a Databricks notebook with the following steps:

- Step 2:** Executed at 10:25 PM (5s). Code: `print("Starting Extracting data from Bronze layer and ingest to silver layer ....")`. Output: `Starting Extracting data from Bronze layer and ingest to silver layer ....`
- Step 3:** Executed at 10:26 PM (<1s). Code: `dbutils.widgets.text("sourcefolder", "Product")` and `dbutils.widgets.text("destinationfolder", "Product")`. The notebook interface shows input fields for `sourcefolder` and `destinationfolder` both containing the value "Product".
- Step 4:** Executed at 10:26 PM (<1s). Code: `var_src_folder = dbutils.widgets.get("sourcefolder")` and `var_tgt_folder = dbutils.widgets.get("destinationfolder")`.
- Step 5:** Executed at 10:26 PM (17s). (Code is partially visible at the bottom of the screenshot).

Save to Silver layer as Delta format and save in Delta Table:

The screenshot shows the continuation of the Databricks notebook with the following steps:

- Step 7:** Executed at 10:33 PM (4s). Code: `# df is your DataFrame`, `# Replace "/mnt/delta/fact_sales_table" with your desired path`, and `df.write.format("delta").mode("overwrite").saveAsTable(f"netflix_data.netflix_schema.{var_tgt_folder}")`. Includes a "See performance (1)" link and an "Optimize" button.
- Step 8:** Executed on Aug 09, 2025 (3s). Code: `df.write.format("delta").mode("overwrite")\`, `.option('overwriteSchema', 'true')\`, `.option("path", f"abfss://silver@azureprojectdatalakegen2.dfs.core.windows.net/{var_tgt_folder}")\`, and `.save()`. Includes a "See performance (1)" link and an "Optimize" button.
- Step 9:** Executed on Aug 09, 2025 (<1s). Code: `print("Completed Extracting data from Bronze layer and ingest to silver layer ....")`. Output: `Completed Extracting data from Bronze layer and ingest to silver layer ....`

# Data Transformation in Silver Layer

## Product Data:

### Product Data

```
▶ Aug 10, 2025 (1s) 3

product_df = spark.read.format("delta")\
    .option("header", "true")\
    .option("inferSchema", "true")\
    .load(f"abfss://silver@azureprojectdatalakegen2.dfs.core.windows.net/Product")

▶ product_df: pyspark.sql.connect.dataframe.DataFrame = [ProductKey: integer, Product: string ... 4 more fields]
```

```
▶ Aug 10, 2025 (2s) 4

product_df.printSchema()
product_df.display()

> See performance \(1\) Optimize

root
|-- ProductKey: integer (nullable = true)
|-- Product: string (nullable = true)
```

```
▶ Aug 10, 2025 (2s) 5

product_df = product_df.drop(col('BackgroundColorFormat'), col('FontColorFormat'))
product_df = product_df.withColumn('StandardCost', regexp_replace('StandardCost', "[,]", "").cast(DoubleType()))
product_df.printSchema()
product_df.display()

> See performance \(1\) Optimize

▶ product_df: pyspark.sql.connect.dataframe.DataFrame = [ProductKey: integer, Product: string ... 4 more fields]

root
|-- ProductKey: integer (nullable = true)
|-- Product: string (nullable = true)
|-- StandardCost: double (nullable = true)
|-- Color: string (nullable = true)
|-- Subcategory: string (nullable = true)
|-- Category: string (nullable = true)
```

Table +

Q Y I □

	<sup>1</sup> / <sub>3</sub> ProductKey	<sup>A</sup> / <sub>C</sub> Product	<sup>1</sup> / <sub>2</sub> StandardCost	<sup>A</sup> / <sub>C</sub> Color	<sup>A</sup> / <sub>C</sub> Subcategory	<sup>A</sup> / <sub>C</sub> Category	
1	210	HL Road Frame - Black, 58	868.63	Black	Road Frames	Components	
2	215	Sport-100 Helmet, Black	12.03	Black	Helmets	Accessories	
3	216	Sport-100 Helmet, Black	13.88	Black	Helmets	Accessories	
4	217	Sport-100 Helmet, Black	13.09	Black	Helmets	Accessories	
5	253	LL Road Frame - Black, 58	176.2	Black	Road Frames	Components	
6	254	LL Road Frame - Black, 58	170.14	Black	Road Frames	Components	

```

region_df.write.format("delta").mode("overwrite")\
    .option("path", f"abfss://{silver}@azureprojectdatalakegen2.dfs.core.windows.net/Region")\
    .save()

```

> [\[1\]](#) See performance (1)

```

    ✓ Aug 10, 2025 (3s) 12
    reseller_df.write.format("delta").mode("overwrite")\
      .option("path", f"abfss://{silver}@azureprojectdatalakegen2.dfs.core.windows.net/Reseller")\
      .save()
    > View See performance (1)

```

```

▶ ✓ Aug 10, 2025 (2s)
sales_df.printSchema()
sales_df.display()
> 📊 See performance \(1\)

```



```

▶ Aug 10, 2025 (2s) 16

sales_df = sales_df\
    .withColumn('UnitPrice', regexp_replace('UnitPrice', "[$,]", "").cast(DoubleType()))\
    .withColumn('Sales', regexp_replace('Sales', "[$,]", "").cast(DoubleType()))\
    .withColumn('Cost', regexp_replace('Cost', "[$,]", "").cast(DoubleType()))\
    .withColumn('OrderDate', to_date(col('OrderDate'), 'dd/MM/yyyy'))

sales_df.printSchema()
sales_df.display()
> See performance \(1\) Optimize

sales_df: pyspark.sql.connect.dataframe.DataFrame = [SalesOrderNumber: string, OrderDate: date ... 8 more fields]
root
|-- SalesOrderNumber: string (nullable = true)
|-- OrderDate: date (nullable = true)
|-- ProductKey: integer (nullable = true)
|-- ResellerKey: integer (nullable = true)
|-- EmployeeKey: integer (nullable = true)
|-- SalesTerritoryKey: integer (nullable = true)
|-- Quantity: integer (nullable = true)
|-- UnitPrice: double (nullable = true)
|-- Sales: double (nullable = true)
|-- Cost: double (nullable = true)

```

## Salesperson Data:

### Salesperson Data

```

▶ Aug 10, 2025 (1s) 19

salesperson_df = spark.read.format("delta")\
    .option("header", "true")\
    .option("inferSchema", "true")\
    .load(f"abfss://silver@azureprojectdatalakegen2.dfs.core.windows.net/Salesperson")

salesperson_df: pyspark.sql.connect.dataframe.DataFrame = [EmployeeKey: integer, EmployeeID: integer ... 3 more fields]

▶ Aug 10, 2025 (3s) 20

salesperson_df.write.format("delta").mode("overwrite")\
    .option("path", f"abfss://silver@azureprojectdatalakegen2.dfs.core.windows.net/Salesperson")\
    .save()
> See performance \(1\) Optimize

```

## Salesperson Region Data:

### SalespersonRegion Data

```

▶ Aug 10, 2025 (1s) 22

salespersonregion_df = spark.read.format("delta")\
    .option("header", "true")\
    .option("inferSchema", "true")\
    .load(f"abfss://silver@azureprojectdatalakegen2.dfs.core.windows.net/SalespersonRegion")

salespersonregion_df: pyspark.sql.connect.dataframe.DataFrame = [EmployeeKey: integer, SalesTerritoryKey: integer]

▶ Aug 10, 2025 (3s) 23

salespersonregion_df.write.format("delta").mode("overwrite")\
    .option("path", f"abfss://silver@azureprojectdatalakegen2.dfs.core.windows.net/SalespersonRegion")\
    .save()
> See performance \(1\) Optimize

```

```

    targets_df = spark.read.format("delta")\
        .option("header", "true")\
        .option("inferSchema", "true")\
        .load(f"abfss://silver@azureprojectdatalakegen2.dfs.core.windows.net/Targets")

```

```
✔ Aug 10, 2025 (3s) 26  
targets_df.printSchema()  
targets_df.display()  
> View See performance (1) Optimize
```

Table  Table  

Q Y Z

	1.1 EmployeeID	1.2 Target	1.3 TargetDate
81	399771412	500000	01-11-2020
82	668991357	500000	01-02-2020
83	668991357	500000	01-03-2020
84	668991357	500000	01-12-2020
85	716374314	500000	01-12-2020
86	841560125	500000	01-03-2020
87	841560125	500000	01-07-2020
88	841560125	500000	01-08-2020
89	841560125	500000	01-09-2020
90	841560125	500000	01-10-2020
91	841560125	500000	01-11-2020
92	841560125	500000	01-12-2020
93	982310417	500000	01-01-2020
94	982310417	500000	01-02-2020
95	982310417	500000	01-03-2020

## Data Aggregation and Serving in Gold Layer

Average cost and Total Cost based on Category and Subcategory:

1. Group by Category and subcategory
2. Use mean function for average and sum function for total
3. Order by Category and Subcategory

The screenshot shows a Databricks notebook with a PySpark query that aggregates data by Category and Subcategory. The query calculates the average and total cost for each subcategory. Below the code, the results are displayed in a table view.

```
product_df_aggregated = product_df.groupBy(["Category", "Subcategory"])\
    .agg(\
        mean(col("StandardCost")).alias("AverageCost"),\
        sum(col("StandardCost")).alias("TotalCost")\
    )\
    .orderBy(["Category", "Subcategory"])\
product_df_aggregated.display()
```

product\_df\_aggregated: pyspark.sql.connect.dataframe.DataFrame = [Category: string, Subcategory: string ... 2 more fields]

	Category	Subcategory	1.2 AverageCost	1.2 TotalCost
1	Accessories	Bike Racks	44.88	44.88
2	Accessories	Bike Stands	59.47	59.47
3	Accessories	Bottles and Cages	2.99	8.97
4	Accessories	Cleaners	2.97	2.97
5	Accessories	Fenders	8.22	8.22
6	Accessories	Helmets	13	117
7	Accessories	Hydration Packs	20.57	20.57
8	Accessories	Lights	12.92	38.76
9	Accessories	Locks	10.31	10.31
10	Accessories	Panniers	51.56	51.56
11	Accessories	Pumps	9.280000000000001	18.560000000000002
12	Accessories	Tires and Tubes	7.288181818181818	80.17

4. Save aggregated table as Delta Lake

The screenshot shows a Databricks notebook with PySpark code to save the original data and the aggregated data as Delta Lake tables. The code uses the write.format("delta").mode("overwrite") method and sets the overwriteSchema option to true.

```
product_df.write.format("delta").mode("overwrite")\
    .option("overwriteSchema", "true")\
    .option("path", f"abfss://gold@azureprojectdatalakegen2.dfs.core.windows.net/Product")\
    .save()

product_df_aggregated.write.format("delta").mode("overwrite")\
    .option("overwriteSchema", "true")\
    .option("path", f"abfss://gold@azureprojectdatalakegen2.dfs.core.windows.net/ProductAggregated")\
    .save()
```

## Top 5 Products as per category

- Use window function from PySpark.
- Define window spec using partition by and order by
- Use dense\_rank() function for appropriate ranking and filter top 5 rows

```

window_spec = Window.partitionBy('Category').orderBy('TotalCost')

top5 = product_df_aggregated.withColumn("Rank", dense_rank().over(window_spec)).filter(col("Rank") <= 5)

top5.show()

```

> [See performance \(1\)](#) Optimize

top5: pyspark.sql.connect.dataframe.DataFrame = [Category: string, Subcategory: string ... 3 more fields]

Category	Subcategory	AverageCost	TotalCost	Rank
Accessories	Cleaners	2.97	2.97	1
Accessories	Fenders	8.22	8.22	2
Accessories	Bottles and Cages	2.99	8.97	3
Accessories	Locks	10.31	10.31	4
Accessories	Pumps	9.280000000000001	18.560000000000002	5
Bikes	Touring Bikes	885.9327272727274	19490.520000000004	1
Bikes	Mountain Bikes	961.8855263157899	36551.650000000016	2
Bikes	Road Bikes	881.709692307693	57311.130000000004	3
Clothing	Socks	3.38	13.52	1
Clothing	Caps	5.953333333333333	17.86	2
Clothing	Vests	23.75	71.25	3
Clothing	Tights	30.929999999999996	92.78999999999999	4
Clothing	Gloves	11.513333333333334	103.62	5
Components	Chains	8.99	8.99	1
Components	Derailleurs	47.275	94.55	2
Components	Brakes	47.29	94.58	3
Components	Headsets	38.660000000000004	115.98	4
Components	Bottom Brackets	40.95333333333333	122.86	5

## Count of cities per Country

```

reseller_df_aggregated = reseller_df.groupBy(col("CountryRegion"))\
    .agg(
        count(col("city")).alias("CityCount")
    )\
    .orderBy(col("CountryRegion"))
reseller_df_aggregated.display()

```

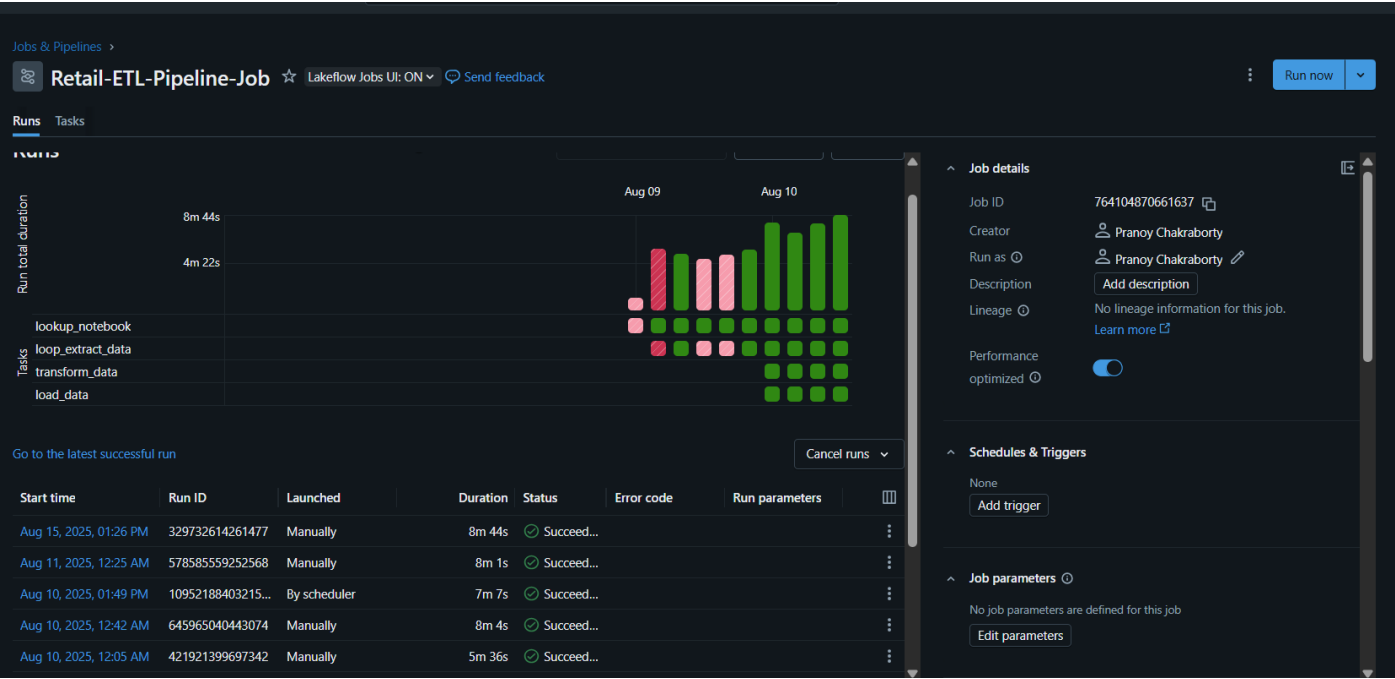
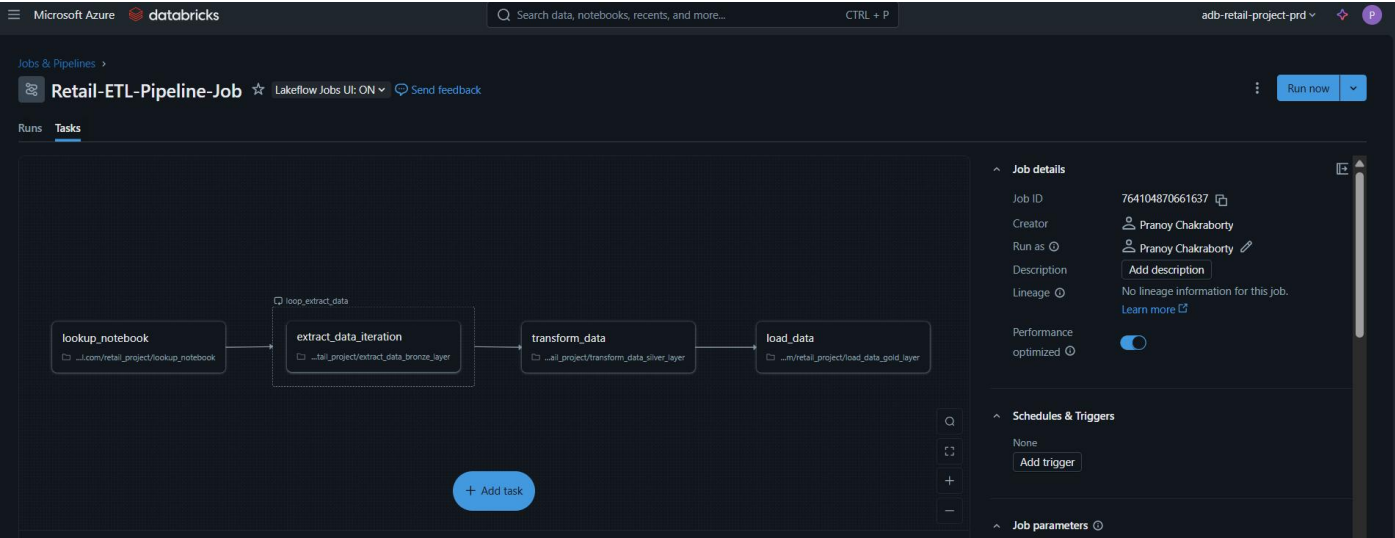
> [See performance \(1\)](#) Optimize

reseller\_df\_aggregated: pyspark.sql.connect.dataframe.DataFrame = [CountryRegion: string, CityCount: long]

	CountryRegion	CityCount
1	Australia	40
2	Canada	114
3	France	40
4	Germany	40
5	United Kingdom	40
6	United States	427

# Retail-ETL-Pipeline-Job

## Lakeflow Jobs UI



### Run details

[View run in Databricks >](#)

Jobs & Pipelines > Retail-ETL-Pipeline-Job >

Retail-ETL-Pipeline-Job run

Lakeflow Jobs UI: ON

Send feedback

Cancel job run

Repair run

Graph

Timeline

List

lookup\_notebook

Succeeded · 47%

↕ 1

../com/retail\_project/lookup\_notebook

loop\_extract\_data

extract\_data\_iteration

Succeeded · 7 scheduled · 5m 30s

../ai\_project/extract\_data\_bronze\_layer

transform\_data

Running · 10s

../ai\_project/transform\_data\_silver\_layer

load\_data

Blocked · 0s

../m/retail\_project/load\_data\_gold\_layer

Job run details

Job ID

764104870661637

Job run ID

231903217517702

Launched

Manually

Started

Aug 22, 2025, 11:50 PM

Ended

-

Duration

6m 36s

Execution time

6m 24s

Queue duration

-

Status

Running

- Cancel

Lineage

0 upstream tables, 7 downstream tables

Performance optimization

Enabled

View run events

Compute

Serverless

Logs

### Run details

[View run in Databricks >](#)

# Delta Tables in Azure Databricks

Catalog

Type to search...

For you All

My organization

adb\_retail\_project\_prd

system

netflix\_catalog

netflix\_data

default

information\_schema

netflix\_schema

product

region

reseller

sales

salesperson

salespersonregion

targets

Delta Shares Received

samples

Legacy

hive\_metastore

File Edit View Run Help Python Tabs: ON Last edit was now

This result is stored as `_sqldf` and can be used in other Python and SQL cells.

Just now (4s) 4

SQL

```
%sql
SELECT
  p.ProductKey,
  p.Product,
  p.Category,
  p.Subcategory,
  s.Sales
FROM
  netflix_data.netflix_schema.product p
  INNER JOIN
  netflix_data.netflix_schema.sales s ON p.ProductKey = s.ProductKey
LIMIT 10;
```

See performance (1) Optimize

`_sqldf: pyspark.sql.connect.dataframe.DataFrame = [ProductKey: integer, Product: string ... 3 more fields]`

Table +

	ProductKey	Product	Category	Subcategory	Sales
1	235	Long-Sleeve Logo Jersey, XL	Clothing	Jerseys	\$57.68
2	351	Mountain-100 Black, 48	Bikes	Mountain Bikes	\$4,049.98
3	348	Mountain-100 Black, 38	Bikes	Mountain Bikes	\$4,049.98
4	232	Long-Sleeve Logo Jersey, L	Clothing	Jerseys	\$57.68
5	292	HL Mountain Frame - Silver, ...	Components	Mountain Frames	\$1,637.4

# Raw data in Bronze Layer in Azure Data Lake

Home > azureprojectdatalakegen2 | Containers >

bronze

Container

Search

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

+ Add Directory

Upload

Refresh

Delete

Copy

Paste

Rename

Acquire le...

bronze

Authentication method: Access key (Switch to Microsoft Entra user account)

Search blobs by prefix (case-sensitive)

Showing all 7 items

<input type="checkbox"/>	Name	Last modified	Access tie
<input type="checkbox"/>	Product	8/15/2025, 1:24:58 PM	
<input type="checkbox"/>	Region	8/15/2025, 1:24:59 PM	
<input type="checkbox"/>	Reseller	8/15/2025, 1:24:59 PM	
<input type="checkbox"/>	Sales	8/15/2025, 1:24:56 PM	
<input type="checkbox"/>	Salesperson	8/15/2025, 1:25:04 PM	
<input type="checkbox"/>	SalespersonRegion	8/15/2025, 1:24:58 PM	
<input type="checkbox"/>	Targets	8/15/2025, 1:25:03 PM	



# Cleaned, Enriched data stored in Silver Layer in Azure Data Lake

Home >

**silver** Container

Search

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

+ Add Directory ↑ Upload ↻ Refresh | 🗑 Delete 📄 Copy 📄 Paste 🔄 Rename 🔗 Acquire lease

silver

Authentication method: Access key ([Switch to Microsoft Entra user account](#))

Search blobs by prefix (case-sensitive)

Showing all 7 items

<input type="checkbox"/>	Name	Last modified	Access tier
<input type="checkbox"/>	Product	8/15/2025, 1:27:44 PM	
<input type="checkbox"/>	Region	8/15/2025, 1:28:24 PM	
<input type="checkbox"/>	Reseller	8/15/2025, 1:28:58 PM	
<input type="checkbox"/>	Sales	8/15/2025, 1:29:28 PM	
<input type="checkbox"/>	Salesperson	8/15/2025, 1:30:08 PM	
<input type="checkbox"/>	SalespersonRegion	8/15/2025, 1:32:03 PM	
<input type="checkbox"/>	Targets	8/15/2025, 1:32:38 PM	

## Data Stored in Delta Format

Home >

**silver** Container

Search

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

+ Add Directory ↑ Upload ↻ Refresh | 🗑 Delete 📄 Copy 📄 Paste 🔄 Rename 🔗 Acquire lease 🔗 Bre

silver > Sales

Authentication method: Access key ([Switch to Microsoft Entra user account](#))

Search blobs by prefix (case-sensitive)

Showing all 5 items

<input type="checkbox"/>	Name	Last modified	Access tier
<input type="checkbox"/>	[.]		
<input type="checkbox"/>	_delta_log	8/15/2025, 1:29:28 PM	
<input type="checkbox"/>	part-00000-22daa571-0ebd-473d-8b7e-94b8ade6d5a5.c000.s...	8/22/2025, 11:57:54 PM	Hot (Inferred)
<input type="checkbox"/>	part-00000-afe58e92-9cff-4807-bee2-29f8c7d75fec.c000.snap...	8/15/2025, 1:29:28 PM	Hot (Inferred)
<input type="checkbox"/>	part-00000-bbc10735-c05a-47a4-b4ca-d7e59247526e.c000.s...	8/15/2025, 1:33:32 PM	Hot (Inferred)
<input type="checkbox"/>	part-00000-cc318d10-b003-4951-ba6a-8c6140d4d074.c000.s...	8/22/2025, 11:54:56 PM	Hot (Inferred)

# Aggregated, Business Ready data Stored in Gold Layer

Home > azureprojectdatalakegen2 | Containers >

gold

Container

Search

◇

<<

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

+ Add Directory

↑ Upload

↻ Refresh

🗑 Delete

📄 Copy

📄 Paste

🏷 Rename

🔗 Acqui

gold

Authentication method: Access key (Switch to Microsoft Entra user account)

Search

Search blobs by prefix (case-sensitive)

Showing all 11 items

<input type="checkbox"/>	Name	Last modified	Access t
<input type="checkbox"/>	ProductAggregated	8/15/2025, 1:34:18 PM	
<input type="checkbox"/>	Region	8/15/2025, 1:34:21 PM	
<input type="checkbox"/>	Reseller	8/15/2025, 1:34:27 PM	
<input type="checkbox"/>	ResellerAggregated	8/15/2025, 1:34:29 PM	
<input type="checkbox"/>	Sales	8/15/2025, 1:34:37 PM	
<input type="checkbox"/>	SalesAggregated	8/15/2025, 1:34:39 PM	
<input type="checkbox"/>	Salesperson	8/15/2025, 1:34:42 PM	
<input type="checkbox"/>	SalespersonRegion	8/15/2025, 1:34:46 PM	
<input type="checkbox"/>	Targets	8/15/2025, 1:34:49 PM	
<input type="checkbox"/>	Top5	8/22/2025, 11:44:42 PM	

Add or remove favorites by pressing Ctrl+Shift+F

Home > azureprojectdatalakegen2 | Containers >

gold

Container

Search

◇

<<

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

+ Add Directory

↑ Upload

↻ Refresh

gold > Top5

Authentication method: Access key (Switch to Micr

Search

Search blobs by prefix (case-sensitive)

Showing all 3 items

<input type="checkbox"/>	Name
<input type="checkbox"/>	[.]
<input type="checkbox"/>	_delta_log
<input checked="" type="checkbox"/>	part-00000-0bbe3959-7e5b-4bbd-9e0e-e07c3f4e4ad0.c000.snappy.parquet
<input type="checkbox"/>	part-00000-8e93df58-cf82-4ac7-971f

Top5/part-00000-0bbe3959-7e5b-4bbd-9e0e-e07c3f4e4ad0.c000.snappy.parquet

Blob

Save

✕ Discard

↓ Download

↻ Refresh

🗑 Delete

↺ Change tier

🔗 Acquire lease

🔗 Break lease

🗨 Give feedback

Overview

Versions

Edit

Generate SAS

Properties

URL

https://azureprojectdatal...🔗

LAST MODIFIED

8/22/2025, 11:44:43 PM

CREATION TIME

8/22/2025, 11:44:43 PM

VERSION ID

-

TYPE

Block blob

SIZE

2.04 KiB

ACCESS TIER

Hot (Inferred)

ACCESS TIER LAST MODIFIED

N/A

ARCHIVE STATUS

-

REHYDRATE PRIORITY

-

SERVER ENCRYPTED

true

ETAG

0x8DDE1A7C4A32A68

VERSION-LEVEL IMMUTABILITY POLICY

Disabled

CACHE-CONTROL

CONTENT-TYPE

application/octet-stream

CONTENT-MD5

CONTENT-ENCODING

CONTENT-LANGUAGE

CONTENT-DISPOSITION

Add or remove favorites by pressing Ctrl+Shift+F