



MicroHack: Microsoft Fabric Lakehouse End-to-End

Erik Eckhardt | Filip Slánička



Agenda

- Introduction to the MicroHack
- Introduction to the Microsoft Fabric Solution
- MicroHack: Microsoft Fabric Lakehouse End 2 End
- Let's Start Hacking

Introducing to the MicroHack



Introducing to the MicroHack

Concept

- Challenge-Based Tutorial Format
 - You'll be actively involved in coding, implementation, and problem-solving.
- What to Expect
 - Real-world scenarios and exercises with direct application of concepts in live environments.
- Hands-On Approach
 - Please read the detailed descriptions provided for each challenge.
 - Carefully review the goals, required actions, and success criteria.
- Support and Collaboration
 - Instructors are here to guide you. Feel free to ask questions and support your fellow participants.
 - A complete end-to-end solution guide with screenshots is also available.
- Outcome
 - By the end of this MicroHack, you'll gain practical experience and new skills that you can directly apply to your projects.

Introducing the Microsoft Fabric solution



Microsoft Data Analytics Evolution





Microsoft Power BI



Azure Synapse Analytics

BI App Live dashboards Visualizations Reports 01001 Datasets
10101

Shared Workspaces

PRO | Premium Per User | Premium Capacity

Cloud-based BI services

Azure Synapse Analytics

Studio

Data Integration Management Monitoring Security

Analytics runtimes

SQL Apache Spark Data Explorer

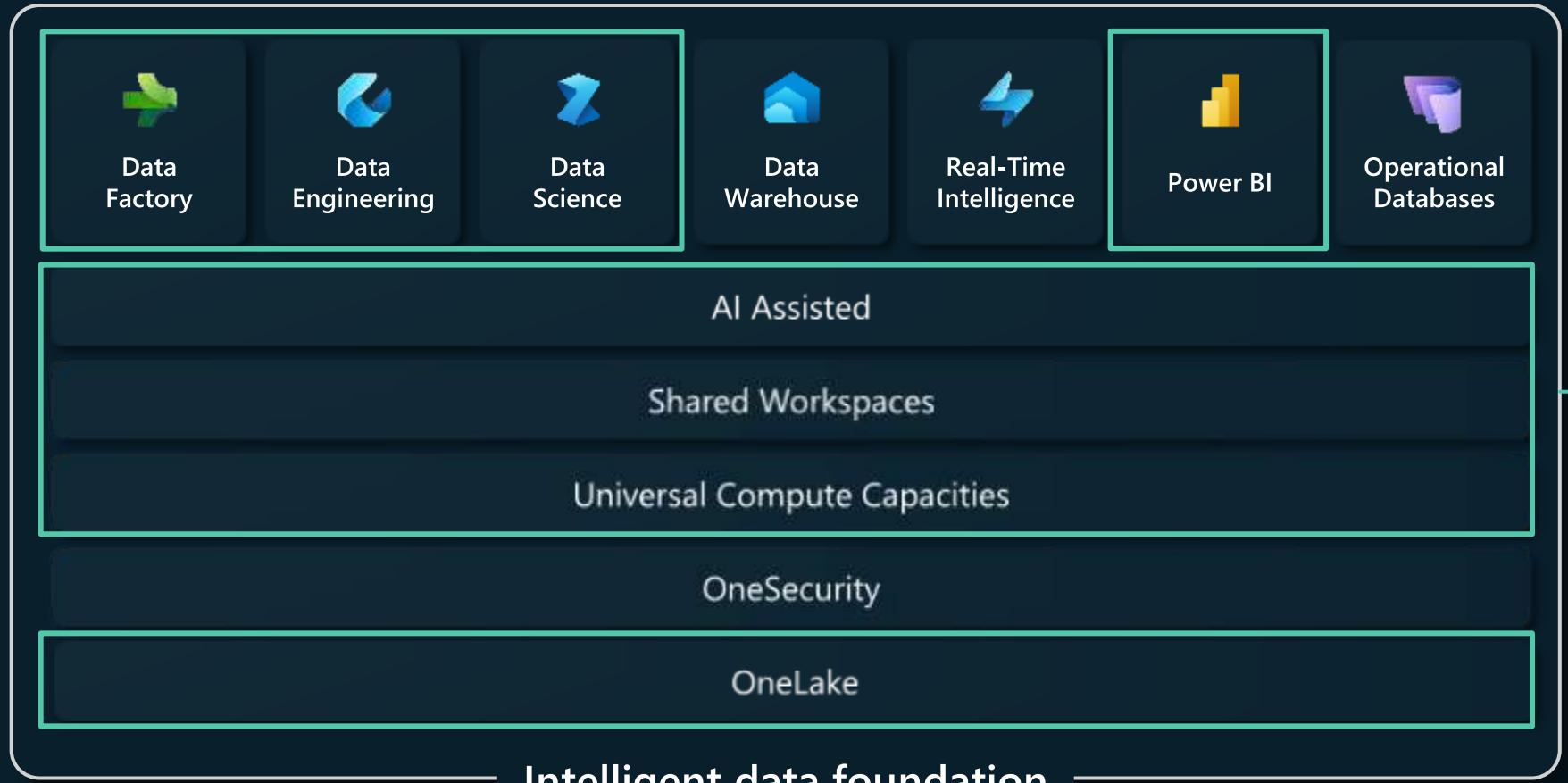
Azure Data Lake Storage Gen2

Cloud-based Analytics services



Microsoft Fabric

The unified data platform for the era of AI



SaaS with single...

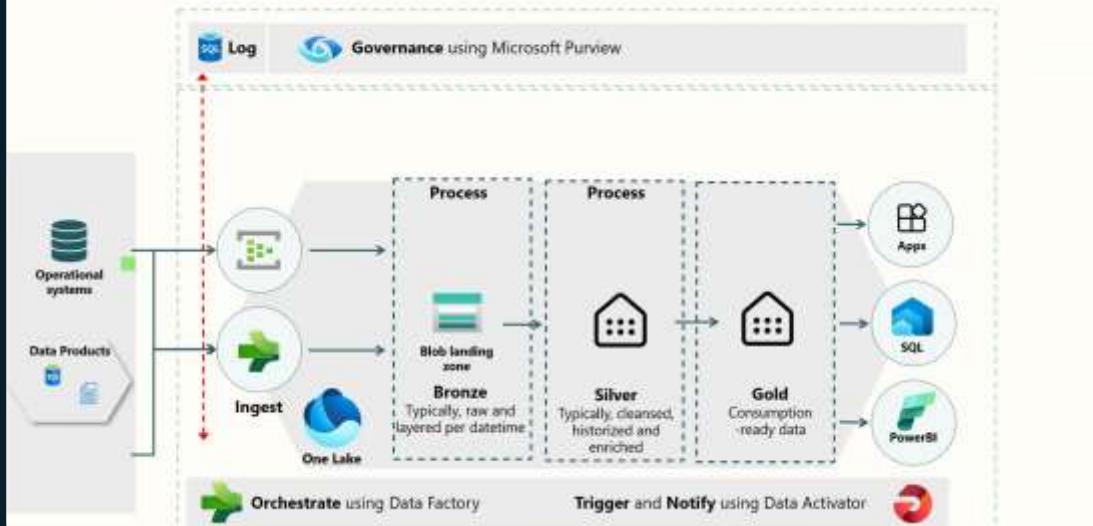
- Onboarding and trials
- Sign-on
- Navigation model
- UX model
- Workspace organization
- Collaboration experience
- Data Lake
- Storage format
- Data copy for all engines
- Security model
- CI/CD
- Monitoring hub
- OneLake Catalog
- Governance & compliance
- Domains



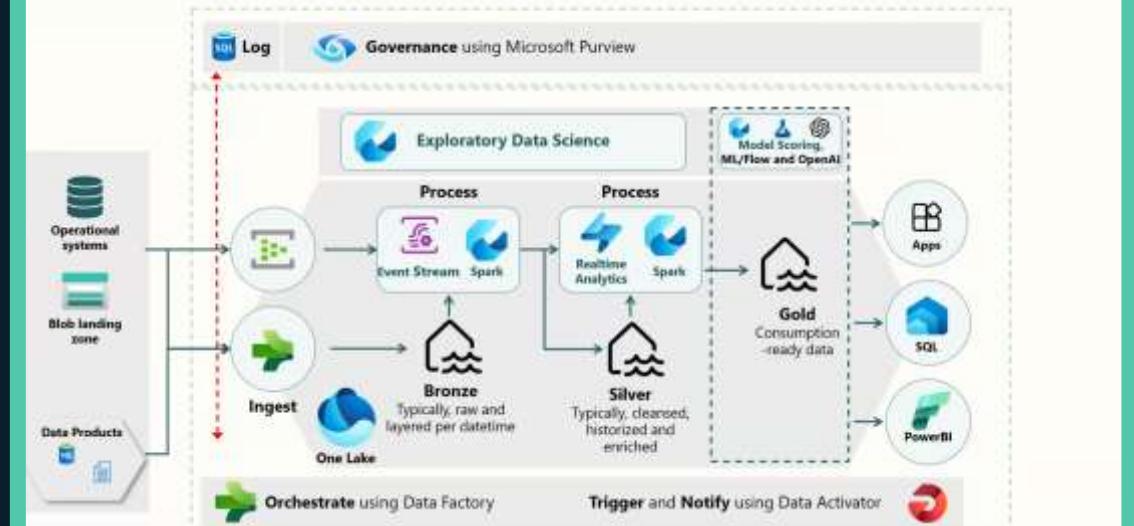
Microsoft Fabric

Data Architecture Patterns

The Warehouse Pattern



The Lakehouse Pattern





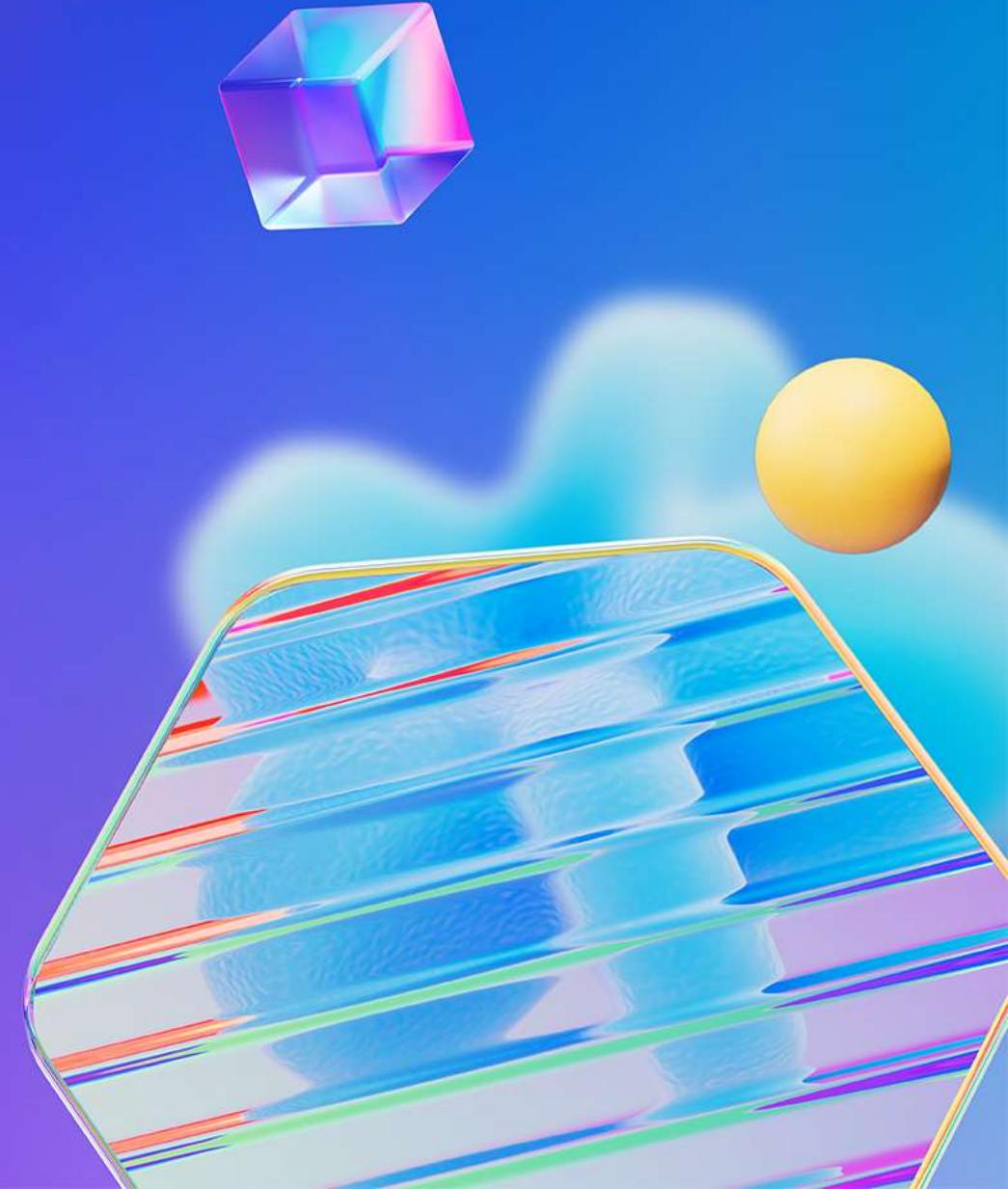
Microsoft Fabric

A lakehouse and a medallion architecture



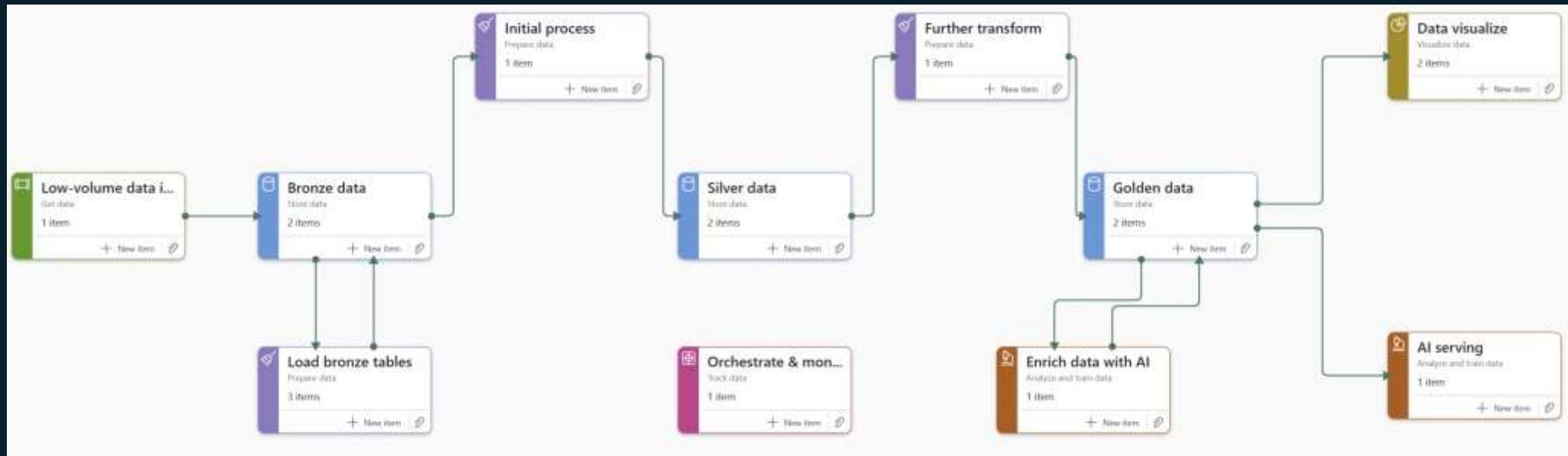
- A lakehouse is an open architecture that combines the best elements of data lakes and data warehouses.
- Lakehouses are enabled by a new system design: implementing similar data structures and data management features to those in a data warehouse directly on top of low-cost cloud storage in open formats with decoupled compute.
- A medallion architecture is a data design pattern used to logically organize data in a lakehouse.
- The goal is to incrementally and progressively improve the structure and quality of data as it flows through each layer of the architecture.
- From a data modeling perspective the Silver Layer has more 3NF write-performant data models, the Gold Layer is more de-normalized and read-optimized data models Kimball style star schema-based or Inmon style Data marts.

MicroHack: Microsoft Fabric Lakehouse End-to-End



MicroHack: Microsoft Fabric Lakehouse End-to-End Context & objectives

- Context
 - Learn how to integrate, transform, and manage data in a lakehouse using Fabric workloads such as Data Factory, Data Engineering, Data Science, and Power BI.
- Objectives
 - Develop a complete data workflow: ingestion, preparation, transformation and serving.



MicroHack: Microsoft Fabric Lakehouse End-to-End Challenges

Important

ch01: Prepare the infrastructure

[Challenge](#) | [Solution](#)

ch02: Ingest raw data

[Challenge](#) | [Solution](#)

ch03: Load *bronze* tables

[Challenge](#) | [Solution](#)

ch04: Transform data using notebooks and Spark clusters

[Challenge](#) | [Solution](#)

ch05: Build *gold* tables and a semantic layer for Power BI reporting

[Challenge](#) | [Solution](#)

ch06: Create a master pipeline for orchestration

[Challenge](#) | [Solution](#)

ch07: Process unstructured data with an LLM

[Challenge](#) | [Solution](#)

ch08: Data Agent: Talk with your data

[Challenge](#) | [Solution](#)

Extra exercises

[Solution](#)

- Focus on implementing **at least the first 6 challenges** today as a minimum out of the 8 major challenges.

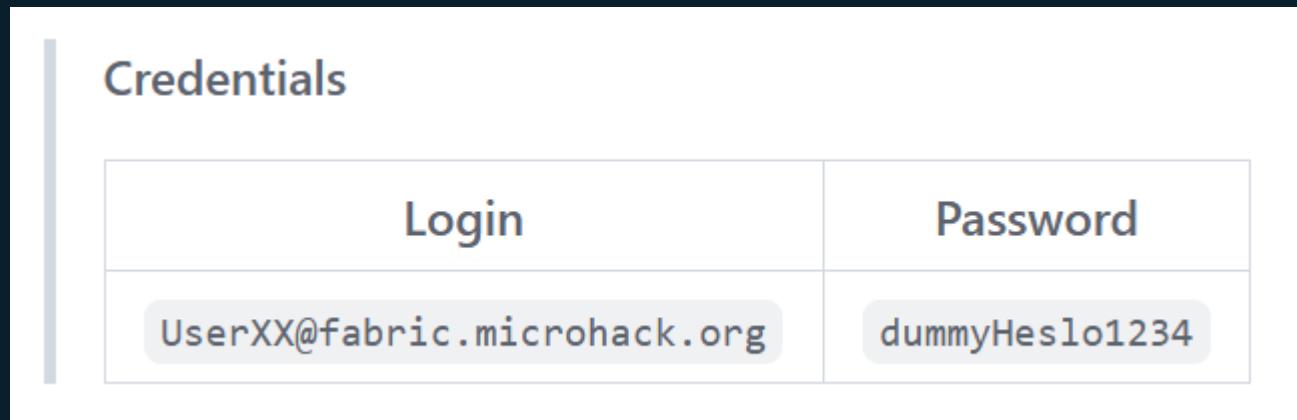
MicroHack: Microsoft Fabric Lakehouse End-to-End

Pre-req. for challenges

 Caution

Before starting the challenges, make sure to go through the step-by-step guide for [New Azure User Setup](#).

- You will receive “XX” now for your credentials. Please remember it!



MicroHack: Microsoft Fabric Lakehouse End-to-End

Ch01: Prepare the infrastructure for Microsoft Fabric

Goal

In this challenge, you will prepare the Microsoft Fabric environment for the upcoming challenges. Each participant should get their own Microsoft account for Azure and Office 365. Using your account, create a Fabric capacity and workspace, attach the capacity to your workspace, and create the first lakehouse for the [bronze layer](#) of the [medallion architecture](#).

Actions

- Deploy MS Fabric Capacity into pre-defined Resource Group "rg-MicroHackFabric" and default Region (West US 3)
 - Name: yoursurnamefabric with size F4
 - e.g. novak0fabric
- Create Workspace in MS Fabric: YourSurname NY Taxi Analysis [DEV]
 - e.g. Novak NY Taxi Analysis [DEV]
- Add your Fabric Capacity to your Workspace.
- Change Spark settings: Use 1 node only
- Create the 1st lakehouse: bronzerawdata

Success Criteria

- MS Fabric Capacity is running.
- Workspace is ready to solve your workloads.
- The 1st Lakehouse and SQL analytics endpoint is deployed and you can connect

NY Taxi Analysis [DEV] ⋮

+ New item New folder Import Migrate

Name	Type	Task
bronzerawdata	Lakehouse	—
bronzerawdata	SQL analytics endpoint	—

Microsoft Fabric ⋮ What platform metrics apply to these resources List CLI commands to manage

Contoso

+ Create Manage view Refresh Export to CSV Open query Assign tags

You are viewing a new version of the browse experience. Click here to access the old experience.

Filter for any field... Subscription equals all Resource Group equals all Location equals all

Name ↑	Type	Resource Group	Location
eec0fabric	Fabric Capacity	rg-MicroHackFabric	West US 3

MicroHack: Microsoft Fabric Lakehouse End-to-End

Ch02: Ingest raw data with data pipelines and shortcuts

Goal

Focus on integrating two data sources from NYC taxi data: one from 2015 and the other from 2023. The table below presents the metaphors we will use to integrate the data:

Actions

- Create Pipeline: Ingest Green Taxi 2015.
 - With Copy Activity to Ingest Data
 - Source: Azure Blob Storage: <https://sfabricmicrohacksw.blob.core.windows.net/>
 - Authentication kind Shared Access Signature (SAS): sv=2024-11-04&ss=b&st=sco&sp=r&tse=2025-11-30T00:36:28Z&st=2025-10-06T15:21:28Z&spr=https&sig=AukZMGIOO%2BdjC%2BAbRmaNpGigBjag2jY0MmASjtTzW%3D
 - File path:
 - Container: taxodata
 - File: green_tripdata_2015-01.parquet
 - File format: Binary
 - Destination : Lakehouse bronzerawdata
 - File path directory: Landing/GreenTaxi2015
 - Format: Binary
- Create Shortcut in your Lakehouse into Files > Landing to 2023 Folder in Azure Data Lake Gen2:
 - <https://sfabricmicrohacksw.dfs.core.windows.net/>
 - Authentication kind Shared Access Signature (SAS): sv=2024-11-04&ss=b&st=sco&sp=r&tse=2025-11-30T00:38:39Z&st=2025-10-06T15:23:39Z&spr=https&sig=HCu0m0ys53EU4M0r0mdmc5KNYNHxhU51WvkR1hRc%3D

Success Criteria

- You have File "green_tripdata_2015-01.parquet" in your Lakehouse: Files > Landing > GreenTaxi2015
 - You can see the content of the file
- You have shortcut to Folder "2023" in your Lakehouse with files "NYC-Taxi-Discounts-Per-Day.csv" and "green_tripdata_2023-01.parquet"
 - You can see the content of both files

Home

Get data New semantic model Open notebook Manage OnLake data access (preview) Update all variables

Explore

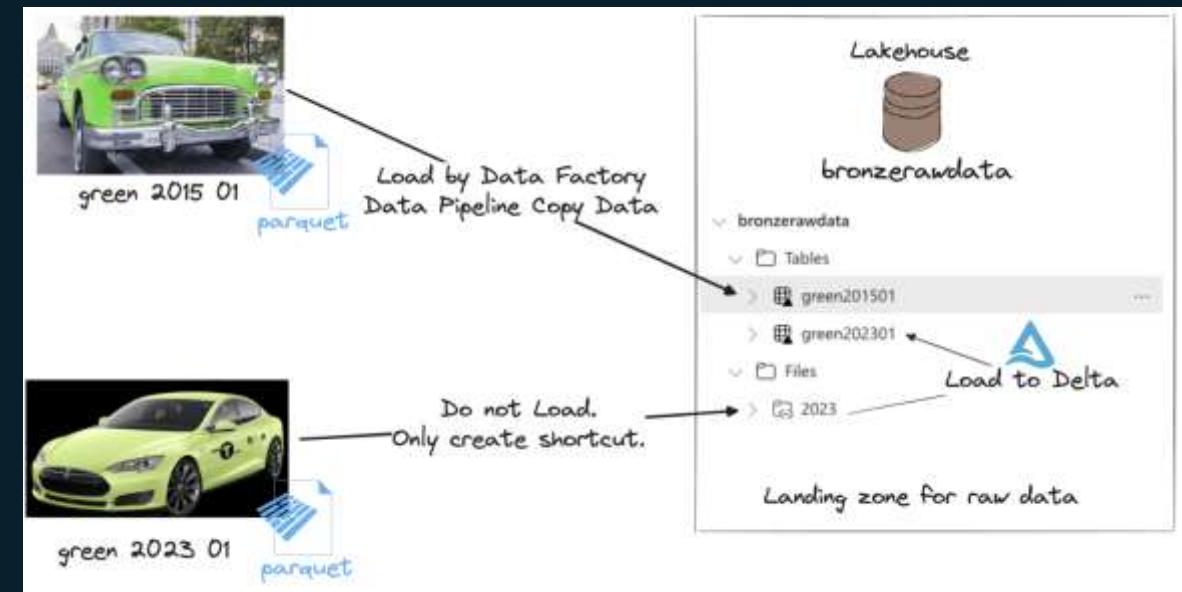
Search tables

bronzerawdata

- Tables
- Files
- Landing
 - 2023
 - GreenTaxi2015

File > Landing > GreenTaxi2015 > green_tripdata_2015-01.parquet (preview)

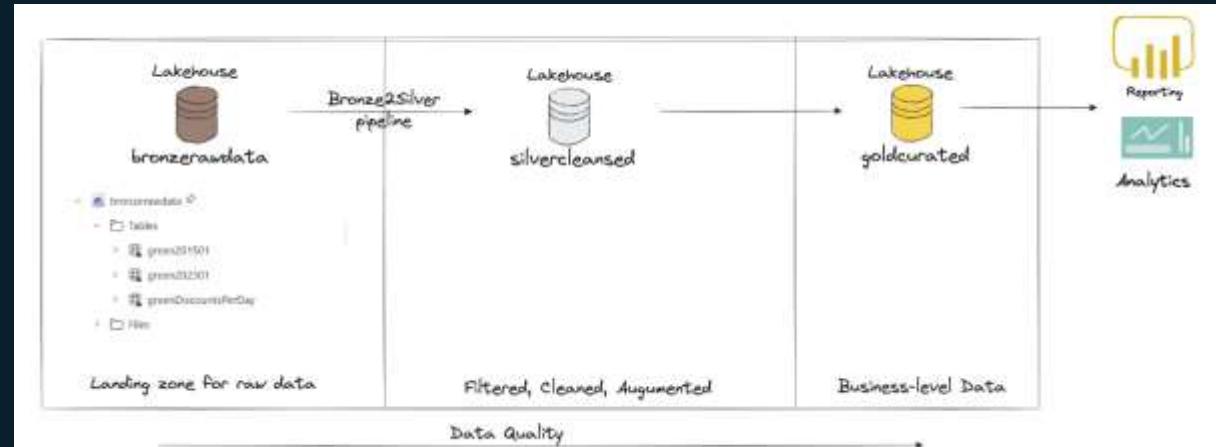
	W. VendorID	lpep_pickup_d	lpep_dropoff_d	mta zone and L.	mta boroughid	mta
1	1	2015-01-01T00..	2015-01-01T00..	N	1	275
2	1	2015-01-01T00..	2015-01-01T00..	N	1	75
3	2	2015-01-01T00..	2015-01-01T00..	N	1	41
4	2	2015-01-01T00..	2015-01-01T00..	N	1	40
5	2	2015-01-01T00..	2015-01-01T00..	N	1	37
6	2	2015-01-01T00..	2015-01-01T00..	N	1	40
7	1	2015-01-01T00..	2015-01-01T00..	N	1	275



MicroHack: Microsoft Fabric Lakehouse End-to-End

Ch03: Load bronze tables

<h2>Goal</h2> <p>In this challenge, you will load three files into three Delta tables in the bronze layer lakehouse. You will use three different load strategies:</p> <ul style="list-style-type: none">1. ETL Developer: Data Factory Pipeline2. Citizen Data Developer: Data Factory Dataflow Gen23. Data Engineer: Notebook <h3>Actions</h3> <ul style="list-style-type: none">• ETL Developer: Data Factory Pipeline<ul style="list-style-type: none">◦ Create a new pipeline: Load Green Taxi 2015<ul style="list-style-type: none">◦ Source:<ul style="list-style-type: none">▪ Connection: bronzerawdata lakehouse▪ Root folder: Files▪ File path: Landing/GreenTaxi2015/green_tripdata_2015-01.parquet▪ File format: Parquet◦ Destination:<ul style="list-style-type: none">▪ Connection: bronzerawdata lakehouse▪ Root folder: Tables▪ Table: green201501 (new table)▪ Table action: Overwrite◦ Create a new pipeline: Load Green Taxi Discounts<ul style="list-style-type: none">◦ Source: bronzerawdata lakehouse<ul style="list-style-type: none">▪ File path: Files > Landing > 2023 > NYC-Taxi-Discounts-Per-Day.csv◦ Destination: bronzerawdata lakehouse◦ Table greenDiscountsPerDay• Citizen Data Developer: Data Factory Dataflow Gen2<ul style="list-style-type: none">◦ Create a new Dataflow Gen2: Load Green Taxi Discounts<ul style="list-style-type: none">◦ Source: bronzerawdata lakehouse▪ File path: Files > Landing > 2023 > NYC-Taxi-Discounts-Per-Day.csv◦ Destination: bronzerawdata lakehouse◦ Table greenDiscountsPerDay• Data Engineer: Notebook<ul style="list-style-type: none">◦ Create a new Notebook: Load Green Taxi 2023 or Import ch03/Load Green Taxi 2023 - Ch03.ipynb<ul style="list-style-type: none">▪ Remove origin lakehouses from Notebook and Add/attach your bronzerawdata (as default) lakehouse◦ Source: Files/Landing/2023/green_tripdata_2023-01.parquet◦ Target: Tables/green202301<ul style="list-style-type: none">▪ Format: delta▪ Mode: overwrites• Check # of rows per delta tables <h3>Success Criteria</h3> <ul style="list-style-type: none">• Table green201501 exists with 1506 498 rows.• Table green202301 exists with 68 231 rows.• Table greenDiscountsPerDay exists with 2 rows. <h4>Explorer</h4> <p>File Items Resources</p> <p>Add data item</p> <p>Home</p> <ul style="list-style-type: none">- bronzerawdata (1)- bronze (1)- green201501 (1)- green202301 (1)- greenDiscountsPerDay (1) <p>111</p> <pre>1. df_201501 = spark.read.format("delta").load("tables/green201501") 2. df_201501 = spark.read.format("delta").load("tables/green201501") 3. df_202301 = spark.read.format("delta").load("tables/green202301") 4. 5. print("green201501 row count:", df_201501.count()) 6. print("green202301 row count:", df_202301.count()) 7. print("greenDiscountsPerDay row count:", df_Discount.count()) 8. 9. # use command processor to run the spark command to read file below 10. # Spark job ID of 0 succeeded: 0 Resources green201501:row count: 1506498 green202301:row count: 68231 greenDiscountsPerDay:row count: 2</pre>



MicroHack: Microsoft Fabric Lakehouse End-to-End

Ch04: Transform data using notebooks and Spark (silver tables)

Goal

In this challenge, you will explore data engineering tasks aimed at transforming raw data into a refined [silver layer](#).

By the end of the challenge, you will have completed the second step, the silver layer, of the [medallion architecture](#):

Actions

- Create the 2nd lakehouse: silvercleansed
- Import [challenges/ch04/Transform bronze to silver - Ch04.ipynb](#) to your workspace and try to complete all Goals / Actions mentioned in notebook
 - Remove origin lakehouses from Notebook and Add/attach your bronzerawdata (as default) lakehouse and silvercleansed lakehouse

Success Criteria

- All success criteria mentioned in the notebook are met
- 6 tables with data in **silvercleansed** lakehouse exist

Explore

Data base: Resources

+ Add data item

- Item

- bronzerawdata
- silvercleansed
 - Tables
 - green201001_cleaned
 - green201001_discounts
 - green201001_revenue
 - green201001_tickets
 - green201001_trips
 - green201001_wheels
 - Files
- File

private ticket records

Table view

#	# recordId	# trip_start_timestamp	# ticketcount	# discounted	# fareCode	# passengerCount	# trip_distance	# trip_duration	# trip
1	2	2015-01-01 00:00:00	100	200	1	1	3.00	10.0	2015-01-01 00:00:00
2	3	2015-01-01 00:00:00	70	140	1	1	0.00	9.4	2015-01-01 00:00:00
3	4	2015-01-01 00:00:00	90	180	1	1	5.00	20.0	2015-01-01 00:00:00
4	5	2015-01-01 00:00:00	80	160	1	1	1.00	10.0	2015-01-01 00:00:00
5	6	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
6	7	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
7	8	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
8	9	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
9	10	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
10	11	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
11	12	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
12	13	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
13	14	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
14	15	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
15	16	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
16	17	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
17	18	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
18	19	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
19	20	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
20	21	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
21	22	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
22	23	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
23	24	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
24	25	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
25	26	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
26	27	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
27	28	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
28	29	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
29	30	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
30	31	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
31	32	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
32	33	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
33	34	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
34	35	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
35	36	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
36	37	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
37	38	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
38	39	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
39	40	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
40	41	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
41	42	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
42	43	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
43	44	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
44	45	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
45	46	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
46	47	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
47	48	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
48	49	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
49	50	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
50	51	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
51	52	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
52	53	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
53	54	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
54	55	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
55	56	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
56	57	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
57	58	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
58	59	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
59	60	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
60	61	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
61	62	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
62	63	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
63	64	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
64	65	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
65	66	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
66	67	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
67	68	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
68	69	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
69	70	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
70	71	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
71	72	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
72	73	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
73	74	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
74	75	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
75	76	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
76	77	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
77	78	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
78	79	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
79	80	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
80	81	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
81	82	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
82	83	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
83	84	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
84	85	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
85	86	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
86	87	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
87	88	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
88	89	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
89	90	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
90	91	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
91	92	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
92	93	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
93	94	2015-01-01 00:00:00	20	40	1	1	10.00	20.0	2015-01-01 00:00:00
94									

MicroHack: Microsoft Fabric Lakehouse End-to-End

Ch05: Build gold tables and a semantic layer for Power BI reporting

Goal

In this challenge, you will aggregate augmented data from a [silver layer](#) into a curated [gold layer](#) for advanced business analysts. Next, you will define semantic layer for end users and create the first Power BI report.

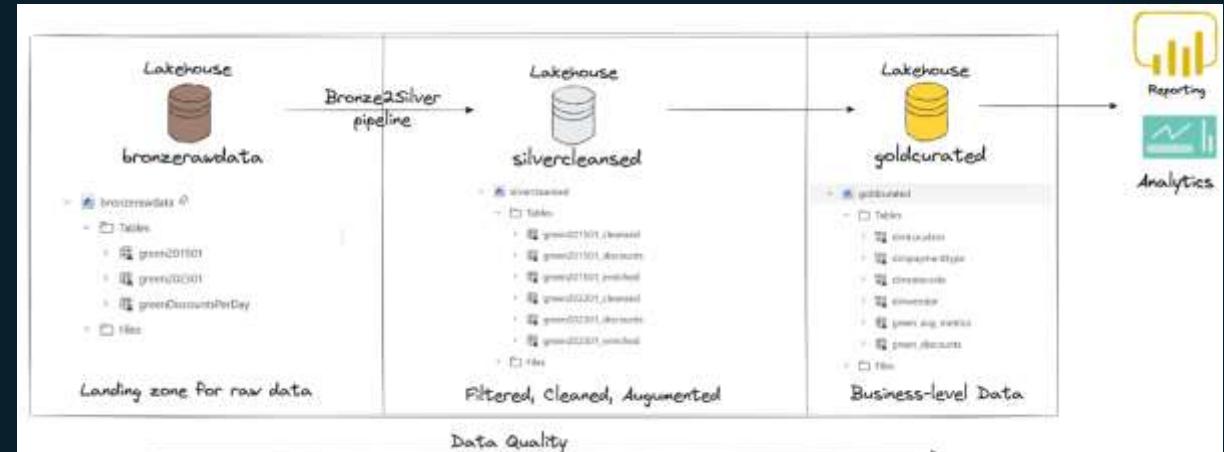
By the end of the challenge, you will have completed the implementation of the [medallion architecture](#) with semantics layer and reports:

Actions

- Create the 3rd lakehouse: goldcurated
- Create Shortcut in your Lakehouse into Tables to pre-created delta tables: dimLocations, dimpaymenttype, dimratecode, dimvendor in Azure Data Lake Gen2: <https://adlsfabricmicrohacks.dfs.core.windows.net/>
 - SAS: sv=2024-11-04&ss=b&sr=sc&sp=r&tse=2025-11-30T00:38:39Z&st=2025-10-06T15:23:39Z&spr=https&sig=HCus0m0ysS3EU4MIOrcmdmc5KNYNNHxhU51WvkR1hRc%3D
- Import [challenges/ch05/Transform silver to gold - Ch05.ipynb](#) to your workspace and try to complete all Goals / Actions mentioned in notebook:
 - Remove origin lakehouses from Notebook and Add/attach your silvercleansed (as default) lakehouse and goldcurated lakehouse
- Build Semantics model GreenTaxi based on 4 dimensions and 2 facts tables with Avg. function and relations
 - dimLocations.LocationID 1:N green_discount.PULocationID
 - dimpaymenttype.payment_type 1:N green_discount.payment_type
 - dimratecode.RatecodeID 1:N green_discount.RateCodeID
 - dimvendor.VendorID 1:N green_discount.VendorID
- dimLocations.LocationID 1:N green_avg_metrics.PULocationID
 - dimvendor.VendorID 1:N green_avg_metrics.VendorID
- Create sample report based on GreenTaxi Semantic model with:
 - 1 Clustered bar chart, 1 Line chart, 1 Azure maps, 1 Narrative and 1 Slicer
- Create SQL View vGetAverageFaresDiscounts based on "Lakehouse SQL analytics endpoint" for business analysts
 - Get the distinct passenger count, total fares, average fares and discount grouped by the passenger count

Success Criteria

- All success criteria mentioned in the notebook are met
- 6 tables with data in goldcurated lakehouse exist
- Semantics model GreenTaxi with 2 Fact tables and 4 Dimensions exists
- Report based on GreenTaxi Semantics model exist
- SQL View vGetAverageFaresDiscounts returns 8 records



MicroHack: Microsoft Fabric Lakehouse End-to-End

Ch06: Create a master pipeline for orchestration

Goal

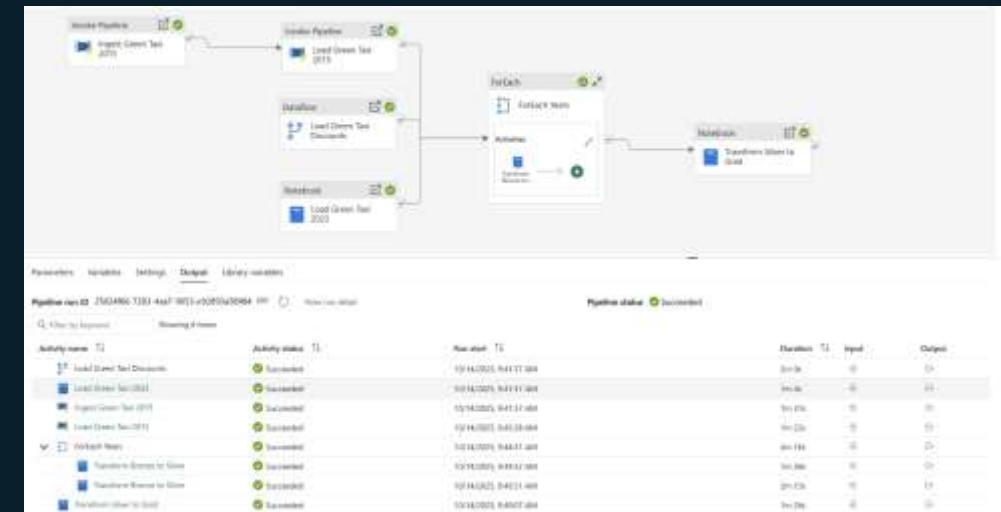
Now, let's focus on orchestration. Considering we have only two tables ingested by two pipelines and processed by three notebooks, imagine a scenario where you need to ingest and process tens or hundreds of tables, files, or data sources and manage dependencies between them. In such cases, the most efficient approach is to build and prioritize a data pipeline for the orchestration workflow.

Actions

- Create master Orchestration Pipeline
 - Add pipeline parameter: Year (Array, Default value = ["2015", "2023"])
 - Build following workflow:
 - Run in parallel:
 - Invoke pipeline: Ingest Green Taxi 2015
 - On Success, run: Invoke Load Green Taxi 2015
 - Run Dataflow: Load Green Taxi Discount
 - Run Notebook: Load Green Taxi 2023
 - On Success, run: ForEach activity: Sequential | Parallel for @pipeline().parameters.Year
 - Run Notebook: Transform Bronze to Silver
 - Add Base parameter p_year binded to ForEach items.
 - On Success: Run Notebook: Transform Silver to Gold
 - Run pipeline

Success Criteria

- Pipeline finishes successfully



MicroHack: Microsoft Fabric Lakehouse End-to-End

Ch07: Process unstructured data with an LLM

Goal

In this challenge, you will focus on processing unstructured data from real user reviews of locations in New York.

With Microsoft Fabric, all business professionals—data engineers, data scientists, developers, and analysts—can gain more value from their data through Generative AI.

There are two options in Microsoft Fabric today:

1. Use [AI services](#) with prebuilt and customizable APIs and models, but you need to develop code to process the output into the final schema:
 - Language Models: gpt-5, gpt-4.1, gpt-4.1-mini, and gpt-4o-mini
 - APIs: Text Analytics, and Azure AI Translator
2. Use the new set of [AI functions](#) for data engineering, which is much easier. Fabric users can take advantage of industry-leading large language models (LLMs) to transform and enrich data with just a single line of code.

By the end of this challenge, you will have a new dataset with sentiment, classification, summarization, translation, key topics, and location extracted from real user reviews.

Tip

It's up to you which option you choose, but this challenge is designed for AI functions.

Actions

- You can create new lakehouse for Sandboxing or use goldcurated
- Upload File [NY_places_customer_reviews.csv](#) to your lakehouse (Files)
- Import [challenges/ch07/User reviews processing with LLM - AI Functions - Ch07.ipynb](#) to your workspace and try to complete all Goals / Actions mentioned in that notebook.
 - Remove origin lakehouses from Notebook and Add/attach your goldcurated (as default) lakehouse

Success Criteria

- All success criteria mentioned in the notebook are met
- Table goldcurated.reviewLocations exists with 265 rows and AI columns

```
1 df = spark.read.format("CSV").option("header", "true").load("goldcurated/reviewslocations LIMIT 1000")
2 display(df)
3 +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 |LocationID|Place|Review|User|Advertisement|AltClassify|AltSummarize|AltTranslationsReview|AltTranslationsSummary|
5 +-----+-----+-----+-----+-----+-----+-----+-----+-----+
6 |1|New York|Tumble Inn|Bocodule|negative|Historic building|User describes a...|Hraniční poklad místní m...|Úžasný postoj k vlastnímu ar...
7 |2|New York|Even in center...|Hubert!!|positive|accommodation|In central NYC, yo...|V centru New Yorku je st...|V centru New Yorku je význa...
8 |3|New York|Pebham Gard...|Ice Cosenti|positive|accommodation|Pebham Gardens is ...|Pebham Gardens je významn...|Pebham Gardens je významn...
9 |4|New York|Alphabet City...|Lulu N|positive|bars|Alphabet City is a ...|Alphabet City představuje j...|Alphabet City je skutečná č...
10 |5|New York|Arden Height...|Anonymous|positive|accommodation|Arden Heights is ...|Arden Heights je domovem ...|Arden Heights je původním ...
11 |6|New York|Beautiful view...|Michael S|mixed|historic building|Beautiful views are...|Kráásné výhledy a zajímav...|Kráásné výhledy a zajímavé h...
12 |7|New York|Astoria is a v...|Current Re...|positive|restaurants|Astoria is a safe ...|Astoria je velmi jediná v...|Astoria je bezpečná, dobré u...
```

```
1 # TRANSLATE input txt to Czech language
2
3 dfp["AltTranslationsReview"] = dfp["Review"].ai.translate("Czech")
4 dfp["AltTranslationsSummarize"] = dfp["Altsummarize"].ai.translate("Czech")
5 display(dfp)
6
```

✓ Command executed in 18 ms (82 ms) by System Administrator on 3:20:05 PM 10/26/25

100% [265/265] [00:07<00:08, 38.36it/s]
100% [265/265] [00:06<00:08, 39.93it/s]

Table view

	LocationID	Place	Review	User	Advertisement	AltClassify	AltSummarize	AltTranslationsReview	AltTranslationsSummary
1	1	New York	Tumble Inn	Bocodule	negative	Historic building	User describes a...	Hraniční poklad místní m...	Úžasný postoj k vlastnímu ar...
2	2	New York	Even in center...	Hubert!!	positive	accommodation	In central NYC, yo...	V centru New Yorku je st...	V centru New Yorku je význa...
3	3	New York	Pebham Gard...	Ice Cosenti	positive	accommodation	Pebham Gardens is ...	Pebham Gardens je významn...	Pebham Gardens je významn...
4	4	New York	Alphabet City...	Lulu N	positive	bars	Alphabet City is a ...	Alphabet City představuje j...	Alphabet City je skutečná č...
5	5	New York	Arden Height...	Anonymous	positive	accommodation	Arden Heights is ...	Arden Heights je domovem ...	Arden Heights je původním ...
6	6	New York	Beautiful view...	Michael S	mixed	historic building	Beautiful views are...	Kráásné výhledy a zajímav...	Kráásné výhledy a zajímavé h...
7	7	New York	Astoria is a v...	Current Re...	positive	restaurants	Astoria is a safe ...	Astoria je velmi jediná v...	Astoria je bezpečná, dobré u...

MicroHack: Microsoft Fabric Lakehouse End-to-End

Ch08: Data Agent: Talk with your data

Goal

In this challenge, you will focus on the Data Agent, which allows you to build your own conversational Q&A systems using generative AI. The Fabric Data Agent sees LLMs to help users interact with their data naturally. It leverages Azure OpenAI Assistant APIs and behaves like an agent: processing user questions, determining the most relevant data source (Lakehouse, Warehouse, Power BI dataset, KQL, databases), and invoking the appropriate tool to generate, validate, and execute queries. This way, even people without technical expertise in AI or a deep understanding of the data structure can receive precise and context-rich answers.

During this challenge, you will define a new Data Agent and add specific instructions, examples, and guidance to fine-tune the Fabric Data Agent. This ensures that responses align with your needs and goals, allowing everyone to engage with data more effectively.

By the end of this challenge, the Data Agent will be ready to be consumed via the Python SDK, Azure AI Foundry, Microsoft Copilot Studio, and Copilot in Power BI.

Actions

- Create new Data Agent
- Add Data sources:
 - : Semantic model: GreenTaxi
 - Tables:
 - dimLocations
 - dimpaymenttype
 - dimratecode
 - dimvendor
 - green_discount
 - Lakehouse: goldcurated
 - Tables:
 - dimLocations
 - reviewLocations
 - Test questions:
 - What were the most common pickup locations for green taxi rides in 2023?
 - On which weekday do people travel from the airport the most?
 - How many rides were in Boston?
 - How many rides were there in 2025?
 - Show me the top 3 districts in New York with the best accommodation.
 - Show me the districts where the most passengers moved from.
 - Show me the districts where most passengers moved from, and what reviews, sentiments, and key topics you know about these districts.
 - Tune Data Agent via Agent instructions (System Prompt) and Data source instructions for above questions
 - Publish Data Agent
 - Consume Data Agent via Copilot (Home) in Power BI or PythonSDK

Success Criteria

- Data Agent exists and is able to answer questions about data in Semantics model and Lakehouse.
- Data Agent is available via Python SDK or Copilot (Home) in Power BI
- Data Agent can answer questions:
 - Question: What were the most common pickup locations for green taxi rides in 2023?
 - Answer: East Harlem North, East Harlem South, Forest Hills, Central Harlem, ...



Let's Start Hacking



MicroHack-Fabric GitHub Repo:
<https://github.com/CZSK-MicroHacks/MicroHack-Fabric>

