

# Data Architecture Build & Market Recommendations for Olist



# Overarching Objectives

## Three Roles

### Data Engineering:

- Build an end-to-end data platform for data processing and BI modelling

### Data Science:

- Enrich the data with sentiment analysis

### Business Intelligence:

- Deliver actionable business insights

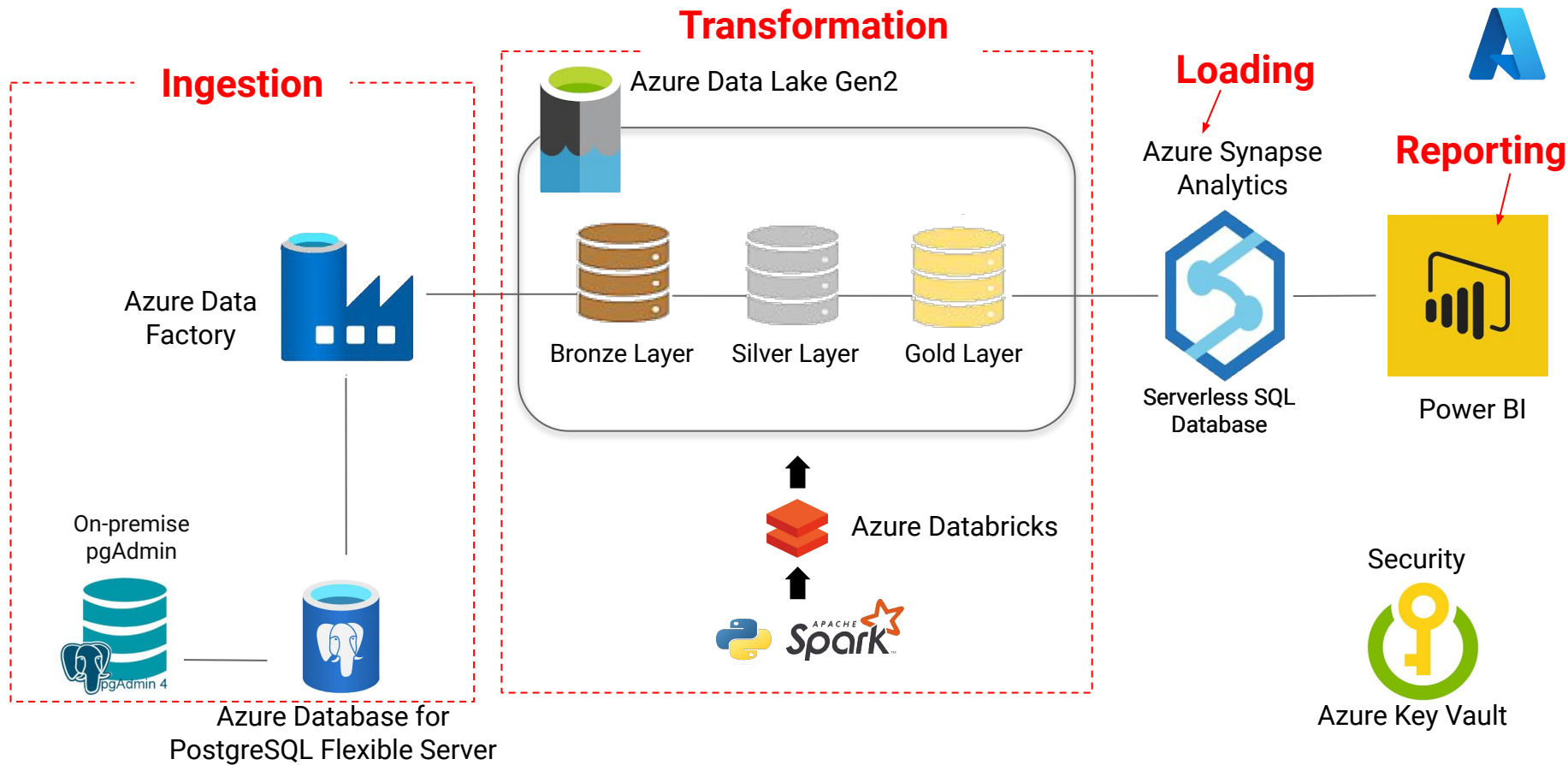
# Introduction to Olist

- Brazil's leading e-commerce marketplace, connecting small businesses to customers nationwide
  - *Average order value: 132 BRL = S\$35 (median salary: S\$2000)*
- **BI goals:**
  - Increase sales turnover
  - Investigation of:
    - Sellers
    - Products
    - Customers



# Data Engineering Objectives

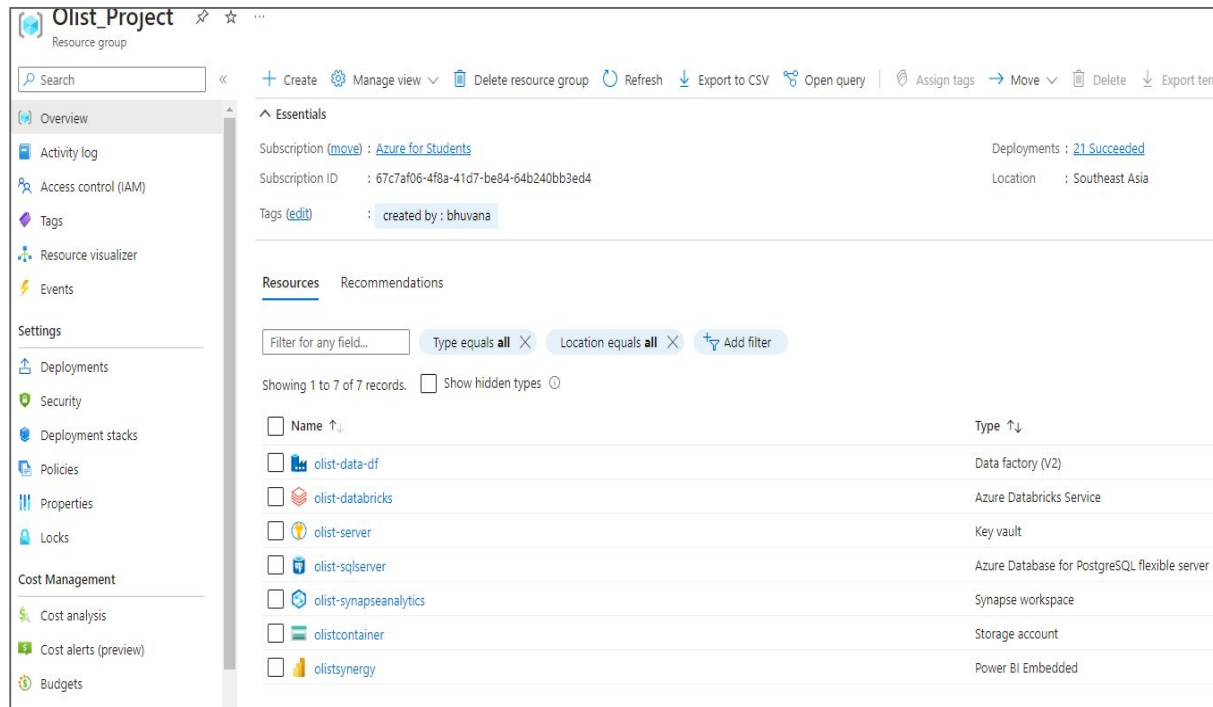
- As a DE team, we gathered project requirements from the BI and data science teams.
- After investigating the use of cloud solutions, we selected Azure.



**Full Azure Cloud-Based Data Flow Architecture**

# Azure Portal – Resource Group Creation and Environmental Setup

- Resource group
- Storage container
- PostgreSQL flexible server
- Azure Data Factory
- Microsoft Integration Runtime
- Azure Databricks
- Azure Synapse Analytics
- Azure Key Vault



# Connection between On-Premise pgAdmin and Azure PostgreSQL Flexible Server

- Setup connection between Azure Database for PostgreSQL flexible server and on-premise pgAdmin

**On-premise pgAdmin**

Object Explorer

- Servers
  - PostgreSQL 16
  - olist-sqlserver**
    - Databases (4)
      - azure\_maintenance
      - azure\_sys
      - olist\_db**
      - postgres

**olist-sqlserver**  
Azure Database for PostgreSQL flexible server

Search

Connect Delete Reset password Restore

POSETTE: An Event for Postgres is happening June 11-13. Join this fr

Essentials

Subscription ([move](#)) : [Azure for Students](#)

Subscription ID : 67c7af06-4f8a-41d7-be84-64b240bb3ed4

Resource group ([move](#)) : [Olist Project](#)

Status : Available

Location : Southeast Asia

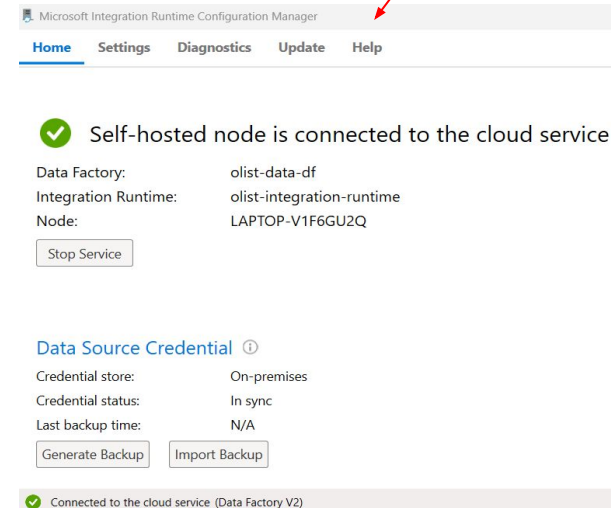
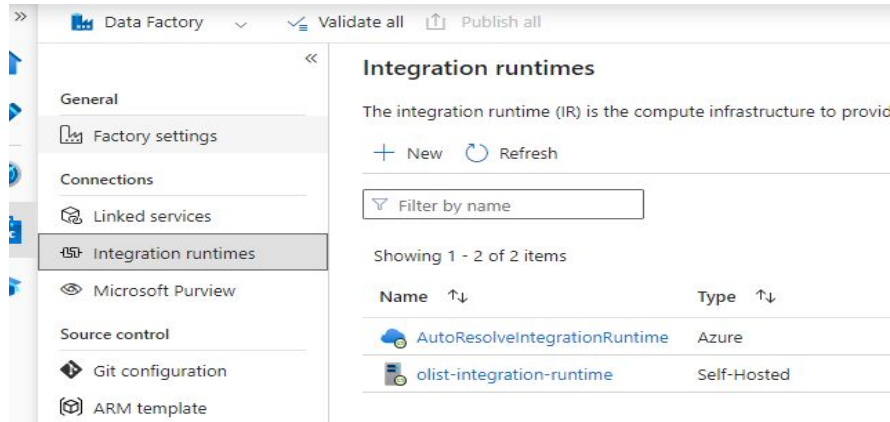
Overview

- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Migration

Settings

- Compute + storage

- There is no relationship between Azure Data Factory and on-premise pgAdmin
- To establish connection we use integration runtimes in ADF
- Self-Hosted Integration Runtime
  - for connecting on-premise pgAdmin
  - Install in our machine to establish connection
- Auto-Resolved Integration Runtime
  - Used to connect to any cloud-based resources



# Installation of Self-Hosted Integration Runtime



# Loading the CSV files into PostgreSQL Database using SQLAlchemy and Accessing through pgAdmin

```
# Retrieve the connection string
connection_string = "postgresql://Username : password @olist-sqlserver.postgres.database.azure.com:5432/olist_db"

# Directory containing CSV files
csv_dir = r"C:\Users\bhu"

# List all CSV files in the directory
csv_files = [f for f in os.listdir(csv_dir) if f.endswith('.csv')]

def process_csv(csv_file):
    table_name = os.path.splitext(csv_file)[0]
    csv_path = os.path.join(csv_dir, csv_file)

    # Create SQLAlchemy engine
    engine = create_engine(connection_string)

    # Process CSV file
    print(f"Processing CSV file: {csv_file}")
    print(f"Processing CSV file: olist_product_category_name_translation.csv")
    print(f"Processing CSV file: olist_sellers_dataset.csv")

    # Create or replace tables
    print(f"Table 'olist_product_category_name_translation' created or replaced successfully in schema 'bronze_layer'")
    print(f"Table 'olist_sellers_dataset' created or replaced successfully in schema 'bronze_layer'")
    print(f"Table 'olist_municipalities_pop' created or replaced successfully in schema 'bronze_layer'")
    print(f"Table 'olist_products_dataset' created or replaced successfully in schema 'bronze_layer'")
    print(f"Table 'olist_order_payments_dataset' created or replaced successfully in schema 'bronze_layer'")
    print(f"Table 'olist_order_reviews_dataset' created or replaced successfully in schema 'bronze_layer'")
    print(f"Table 'olist_customers_dataset' created or replaced successfully in schema 'bronze_layer'")
    print(f"Table 'olist_order_items_dataset' created or replaced successfully in schema 'bronze_layer'")
    print(f"Table 'olist_orders_dataset' created or replaced successfully in schema 'bronze_layer'")
    print(f"Table 'olist_geolocation_dataset' created or replaced successfully in schema 'bronze_layer'")

    # Process results
    print("Processing results:")
    print("ThreadPoolExecutor shutdown complete")
```

# Tables in the Olist\_db Database

The screenshot displays a database management interface for the `olist_db` database. The left pane shows the database structure, and the right pane shows the list of tables.

**Left Pane (Database Structure):**

- olist\_db
  - > Casts
  - > Catalogs
  - > Event Triggers
  - > Extensions
  - > Foreign Data Wrappers
  - > Languages
  - > Publications
  - ✓ Schemas (2)
    - > bronze\_layer
    - > public

**Right Pane (Tables):**

- > 1.3 Sequences
- ✓ Tables (10)
  - > olist\_customers\_dataset
  - > olist\_geolocation\_dataset
  - > olist\_municipalities\_pop
  - > olist\_order\_items\_dataset
  - > olist\_order\_payments\_dataset
  - > olist\_order\_reviews\_dataset
  - > olist\_orders\_dataset
  - > olist\_product\_category\_name\_translation
  - > olist\_products\_dataset
  - > olist\_sellers\_dataset

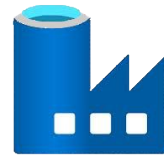
A red arrow points from the `public` schema in the left pane to the table list in the right pane.



On-premise  
Database



Azure Database for  
PostgreSQL Flexible Server



Azure Data Factory



Azure Data Lake Gen2

home > olistcontainer

## olistcontainer | Containers

Storage account

Search

- Overview
- Activity log
- Tags
- Diagnose and solve problems
- Access Control (IAM)
- Data migration
- Events

+ Container

Search containers by

Name
<input type="checkbox"/> \$logs
<input type="checkbox"/> olist-bronze
<input type="checkbox"/> olist-gold
<input type="checkbox"/> olist-silver

Name

<input type="checkbox"/> [..]
<input type="checkbox"/> olist_customers_dataset
<input type="checkbox"/> olist_geolocation_dataset
<input type="checkbox"/> olist_municipalities_pop
<input type="checkbox"/> olist_order_items_dataset
<input type="checkbox"/> olist_order_payments_dataset
<input type="checkbox"/> olist_order_reviews_dataset
<input type="checkbox"/> olist_orders_dataset
<input type="checkbox"/> olist_product_category_name_trar
<input type="checkbox"/> olist_products_dataset
<input type="checkbox"/> olist_sellers_dataset

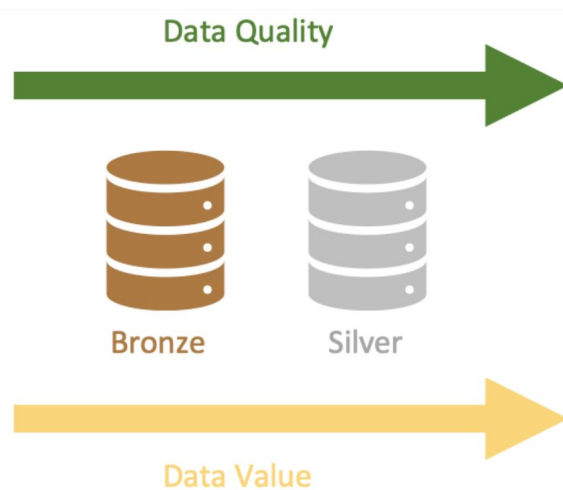
Name

<input type="checkbox"/> [..]
<input type="checkbox"/> olist_customers_dataset.parquet

# Ingestion of Dataset into the Bronze Layer

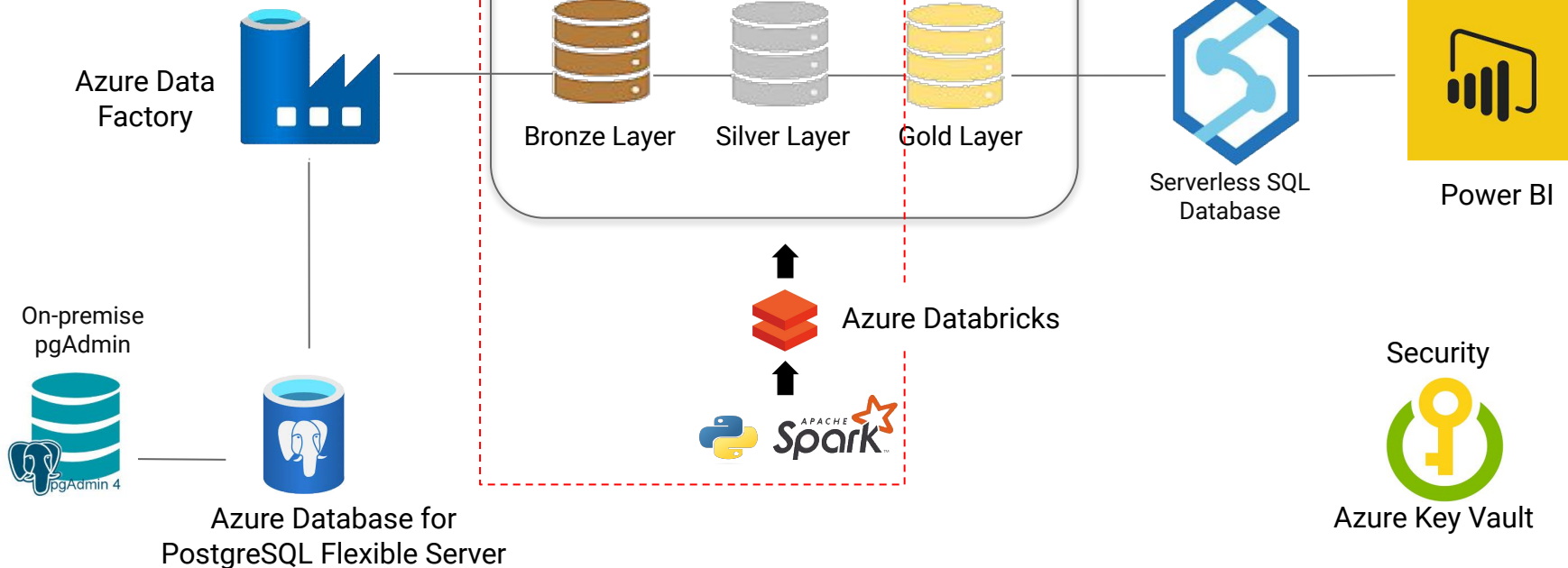


# DATA TRANSFORMATION #1





GitHub



## Full Azure Cloud-Based Data Flow Architecture

# Compute Cluster

The screenshot displays the Microsoft Azure Databricks web interface. At the top, the 'Microsoft Azure' logo is on the left, and the 'databricks' logo is on the right, enclosed in a yellow rectangular highlight. Below the logos is a dark sidebar with navigation options: '+ New', 'Workspace', 'Recents', 'Catalog', 'Workflows', 'Compute' (which is highlighted), and 'SQL'. The main content area is titled 'Compute' and features three tabs: 'All-purpose compute' (selected), 'Job compute', and 'SQL wareho'. Below the tabs is a search bar with the placeholder text 'Filter compute you have access to'. At the bottom, a table lists compute clusters with columns for 'State' and 'Name'. One cluster is visible, named 'Olist\_transformation', with a 'Running' state icon.

Microsoft Azure databricks

+ New

Workspace

Recents

Catalog

Workflows

Compute

SQL

## Compute

All-purpose compute Job compute SQL wareho

Filter compute you have access to

State	Name
Running	Olist_transformation

# Storage Mount

The screenshot displays a Databricks workspace interface. On the left, the 'Workspace' sidebar shows a file tree with the following items: 'gold\_transformation', 'olist-silver-transform', 'olist-transform2', 'Reviews Text Translation', and 'storage mount'. The 'storage mount' file is highlighted with a red rectangular box. The main editor area shows a Python script with the following code:

```
configs = {  
    "fs.azure.account.auth.type": "CustomAccessToken",  
    "fs.azure.account.custom.token.provider.class": spark.conf.get("spark.databricks.passthrough.adls.gen2.tokenProviderClassName")  
}  
  
# Check if the directory is already mounted  
is_mounted = False  
for mount in dbutils.fs.mounts():  
    if mount.mountPoint == "/mnt/olist-bronze":  
        is_mounted = True  
        break  
  
# Unmount the directory if it is already mounted  
if is_mounted:  
    dbutils.fs.unmount("/mnt/olist-bronze")  
  
# Mount the directory  
dbutils.fs.mount(  
    source = "abfss://olist-bronze@olistcontainer.dfs.core.windows.net",  
    mount_point = "/mnt/olist-bronze",  
    extra_configs = configs)  
  
/mnt/olist-bronze has been unmounted.  
True
```

Below the script, the execution output shows the message: `/mnt/olist-bronze has been unmounted.` followed by `True`. The top of the interface includes a header with the file name 'storage mount', a language dropdown set to 'Python', a star icon, and a menu bar with 'File', 'Edit', 'View', 'Run', and 'Help'. A status bar at the bottom indicates 'Last edit was 6 days ago' and 'New cell UI: ON'.

# Data Ingestion

- **Nine CSV files:**

- Orders
- Customers
- Sellers
- Geolocations
- Order Items
- Order Reviews
- Order Payments
- Products
- Product Category Name Translation

- **10th CSV file:**

- Census 2022 (*IBGE*)  
<https://sidra.ibge.gov.br/pesquisa/censo-demografico/demografico-2022/primeiros-resultados-populacao-e-domicilios>



▶ ✓ 5 days ago (<1s)

2

```
input_path_trans= '/mnt/olist-bronze/bronze_layer/olist_product_category_name_translation/olist_product_category_name_translation.parquet'
input_path_prod = '/mnt/olist-bronze/bronze_layer/olist_products_dataset/olist_products_dataset.parquet'
input_path_items = '/mnt/olist-bronze/bronze_layer/olist_order_items_dataset/olist_order_items_dataset.parquet'
input_path_orders = '/mnt/olist-bronze/bronze_layer/olist_orders_dataset/olist_orders_dataset.parquet'
input_path_payments = '/mnt/olist-bronze/bronze_layer/olist_order_payments_dataset/olist_order_payments_dataset.parquet'
input_path_customers = '/mnt/olist-bronze/bronze_layer/olist_customers_dataset/olist_customers_dataset.parquet'
input_path_seller
input_path_review
input_path_geo =
input_path_censu
```

▶ ✓ 4 days ago (2s)

4

```
columns = {0: "product_category_name", 1: "product_category_name_english"}
product_trans_df = pd.DataFrame(spark.read.format('parquet').load(input_path_trans,header=True).collect())
product_trans_df.rename(columns=columns, inplace=True)
print(product_trans_df.head())
print(product_trans_df.info())
```

▶ (2) Spark Jobs

	product_category_name	product_category_name_english
0	beleza_saude	health_beauty
1	informatica_acessorios	computers_accessories

# Data Ingestion

# Data Cleaning I

- Null/NaN values (e.g., review comments, product attributes)
- Data type constraints (e.g., timestamps, integers)
- Inaccurate spelling in column names (e.g., **“lenght”**)
- Non-standard English or typos in translations of product category names (e.g., **“telephony”**, **“home confort”**)
- Portuguese accented characters in city names (e.g., **â, ç**)

# Data Cleaning II

## 1. Out of range data

```
# Drop latitude & longitude outliers (relative to Brazil's range of coordinates)
geolocations_df3 = geolocations_converted.copy()

geolocations_df3 = geolocations_df3[geolocations_df3["geolocation_lat"] <= 5.288685]
assert geolocations_df3["geolocation_lat"].max() <= 5.288685

geolocations_df3 = geolocations_df3[geolocations_df3["geolocation_lat"] >= -33.798533]
assert geolocations_df3["geolocation_lat"].min() >= -33.798533

geolocations_df3 = geolocations_df3[geolocations_df3["geolocation_lng"] <= -34.703311]
assert geolocations_df3["geolocation_lng"].max() <= -34.703311

geolocations_df3 = geolocations_df3[geolocations_df3["geolocation_lng"] >= -73.968899]
assert geolocations_df3["geolocation_lng"].min() >= -73.968899
```

# Data Cleaning II

## 2. Missing • Time

```
# Fill missing values for 'delivered_orders'
for column in columns_to_fill:
    if column == 'order_approved':
        delivered_orders.loc[delivered_orders[column].isnull(), 'order_approved'] = 'D'
    elif column == 'order_delivered':
        delivered_orders.loc[delivered_orders[column].isnull(), 'order_delivered'] = 'D'
    elif column == 'order_purchased':
        delivered_orders.loc[delivered_orders[column].isnull(), 'order_purchased'] = 'D'
```

```
# Iteratively repeat the procedure until no further missing zipcodes are found
while zip_cust_list:
    geolocations_df5 = geolocations_df6

    subs_zip3 = []
    for missing_zip3 in zip_cust_list:
        subs_zip3.append(closest(lst2, missing_zip3))

    cust_dict2 = dict(zip(subs_zip3, zip_cust_list))

    cust_zip_df2 = geolocations_df5[geolocations_df5["geolocation_zip_code_prefix"].isin(subs_zip3)]
    cust_zip_df3 = cust_zip_df2.copy()
    cust_zip_df3["geolocation_zip_code_prefix"] = cust_zip_df3["geolocation_zip_code_prefix"].map(cust_dict2)

    geolocations_df6 = pd.concat([geolocations_df5, cust_zip_df3])
    geolocations_df6.reset_index(drop=True)
    print("Looping...")

    zip_cust_list = list(set(customers_df4["customer_zip_code_prefix"]).difference(set(geolocations_df6["geolocation_zip_code_prefix"])))
```

# Data Cleaning II

```
# Replacing unstandardised city names in Sellers dataset with official names, matched by zip code
```

```
sellers_subset3 = sellers_subset
```

```
sellers_df3 = sellers_df2.copy()
```

```
sellers_df3["city"].update(sellers_subset3["city"])
```

```
sellers_df3.rename(columns={"city": "official_city", "zip": "zip_code"})
```

```
# Standardisation #2 Replacing 230 unstandardised city names with official demographic city names,
```

```
# then testing with assert statements
```

```
'conservatoria': "valenca",
```

```
'itamira': "apora",
```

```
'quatituba': "itueta",
```

```
'santo amaro de campos': "campos dos goytacazes",
```

```
'travessao': "campos dos goytacazes"}
```

```
customers_df4.replace({"official_city": city_dict}, inplace=True)
```

```
assert customers_df4[customers_df4["official_city"] == "abranetes"].empty == True
```

```
assert customers_df4[customers_df4["official_city"] == "adhemar de barros"].empty == True
```

res

- Source

- 

- Two

- I. Zip code matching with Geolocations' official cities
- II. Obtaining official city name from Google Maps

# Data Cleaning II

## 4. Duplicate values

- >1,000,000 Geolocations records, but only 19,000 unique zip code prefixes

```
# Group records by zip_code_prefix, getting mean coordinates & modal city and state names
# Then reset index to make zip_code_prefix a column
geolocations_df4 = geolocations_df3.copy()
✓ summaries = {"geolocation_lat": "mean",
               "geolocation_lng": "mean",
               "geolocation_city": pd.Series.mode,
               "geolocation_state": pd.Series.mode}
geolocations_df4 = geolocations_df4.groupby(by="geolocation_zip_code_prefix").agg(summaries)
```



# Entity Relationship Diagram



```
# Convert Pandas DataFrame to Spark DataFrame
```

```
order_items_spark_df = spark.createDataFrame(products_df4)
```


```
order_items_spark_df = spark.createDataFrame(order_items_df2)
```

```
# Write Spark DataFrame to delta
```

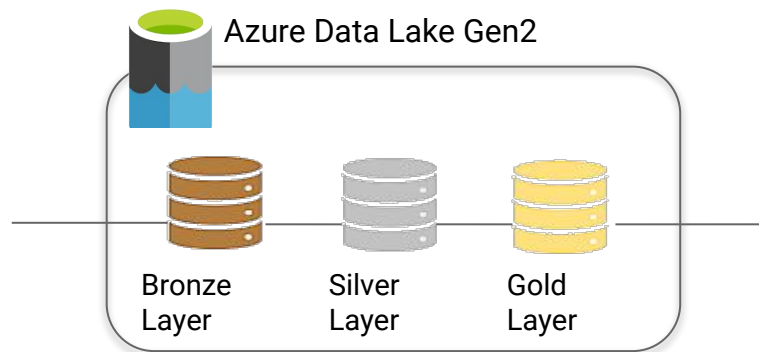
```
order_items_spark_df.write.format("delta").mode("overwrite").option("header", "true").save("dbfs:/mnt/olist-silver/products")
```

```
order_items_spark_df.write.format("delta").mode("overwrite").option("header", "true").save("dbfs:/mnt/olist-silver/order_items")
```

► (4) Spark Jobs

►  order\_items\_spark\_df: pyspark.sql.dataframe.DataFrame = [product\_id: string, product\_category\_name: string ... 8 more fields]

- **Delta format:**
  - Better data management



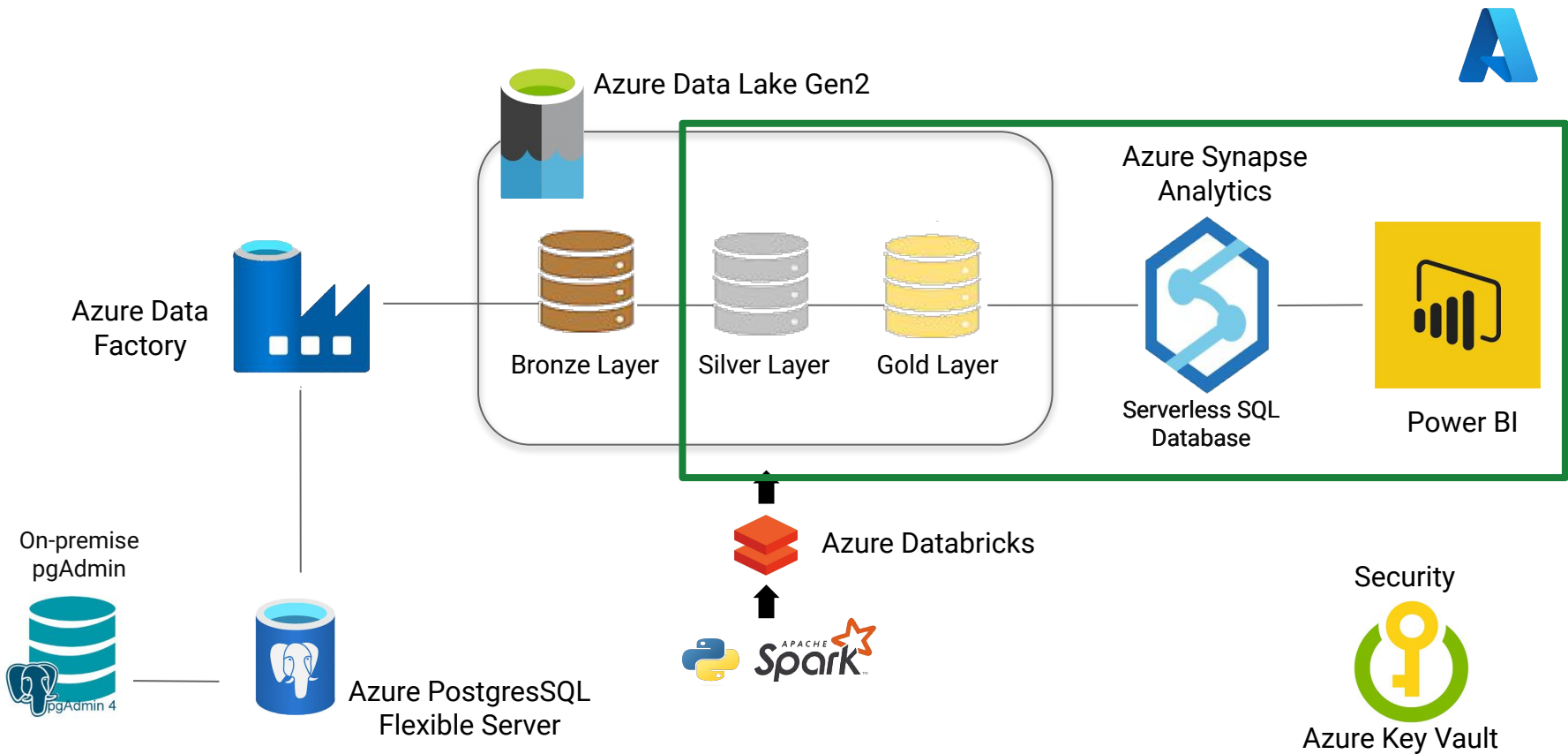
# Data Loading





# DATA TRANSFORMATION #2










## Full Azure Cloud-Based Data Flow Architecture

# Silver to Gold Transformation

Created new workspace for Gold transformation

Workspace >				
Shared ☆				
Name 	Type	Owner	Created at	
 gold_transformation	Notebook	DORAIRAJ BHUVANESHWARI	2024-05-01 23:22:58	
 olist-silver-transform	Notebook	DORAIRAJ BHUVANESHWARI	2024-05-03 11:58:54	

# Extraction and Ingestion of Silver Data

```
# Access the dataframe by name
census_df = dataframes[0]
customers_df = dataframes[1]
geolocations_df = dataframes[2]
order_items_df = dataframes[3]
order_payments_df = dataframes[4]
reviews_full_df = dataframes[5]
orders_df = dataframes[6]
products_df = dataframes[7]
sellers_df = dataframes[8]

census_df = census_df.toPandas()
customers_df = customers_df.toPandas()
geolocations_df = geolocations_df.toPandas()
order_items_df = order_items_df.toPandas()
order_payments_df = order_payments_df.toPandas()
reviews_full_df = reviews_full_df.toPandas()
orders_df = orders_df.toPandas()
products_df = products_df.toPandas()
sellers_df = sellers_df.toPandas()
```

**Connect all the  
silver files data  
(Delta format)  
from Data Lake  
to Databricks,  
and block convert  
them from  
Spark DataFrame to  
Pandas DataFrame**

# Translation of Text Reviews



Azure text  
translation service  
API

Portuguese to English

# A Section of Translation Code

```
36
37     tracker = []
38     body = []
39     if not is_null_or_empty(row['review_comment_title']):
40         tracker.append('review_comment_title')
41         body.append({
42             'text': row['review_comment_title']
43         })
44     if not is_null_or_empty(row['review_comment_message']):
45         tracker.append('review_comment_message')
46         body.append({
47             'text': row['review_comment_message']
48         })
49
50     try:
51         response = requests.post(constructed_url, params=params, headers=headers, json=body)
52         response.raise_for_status()
53         response_content = json.loads(response.text)
54         review_comment_title_translated = None
55         review_comment_title_index = find_index_in_list(tracker, 'review_comment_title')
56         review_comment_message_tr_translated = None
57         review_comment_message_tr_index = find_index_in_list(tracker, 'review_comment_message')
58         if review_comment_title_index != -1:
59             review_comment_title_translated = response_content[review_comment_title_index]['translations'][0]['text']
60         if review_comment_message_tr_index != -1:
61             review_comment_message_tr_translated = response_content[review_comment_message_tr_index]['translations'][0]['text']
62
63     return {"status_code": response.status_code if 'response' in locals() else None,
64           "processed_at": datetime.now(timezone.utc),
```

# Translation Code

1. Point to location of source data and load into a dataframe
2. Create source and destination tables to input data
3. Process data using Azure Translator API -  
translate the review comment column
4. Handle the API requests and responses in chunks of 5 reviews per time  
and throw error messages if any
5. Save the destination table

# Reducing Data Size to a Sample of 1000

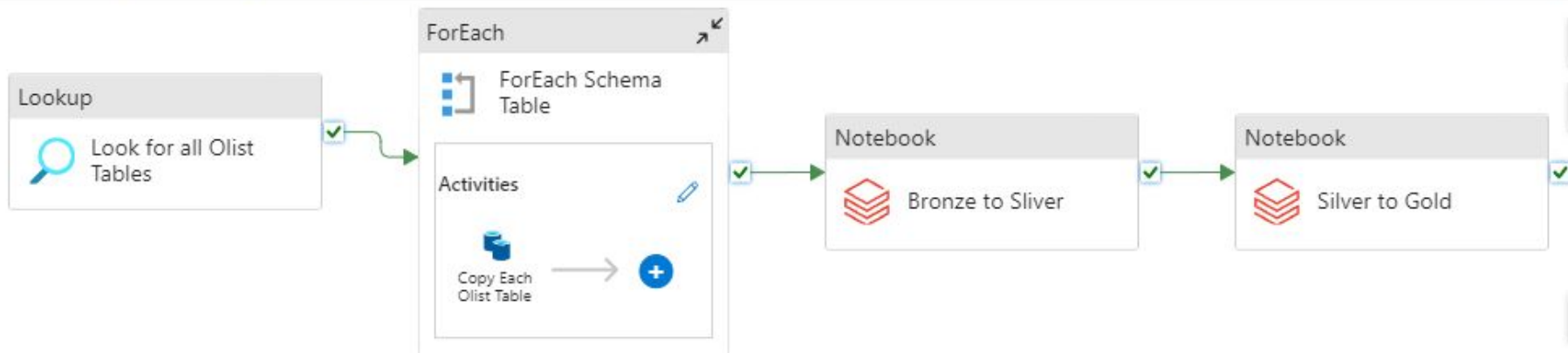
```
27
28 # Randomly sample 1000 rows from reviews_text
29 reviews_text = reviews_text.sample(n=1000, random_state=42)
30
```

A <sup>B</sup> <sub>C</sub> review_comment_message	A <sup>B</sup> <sub>C</sub> review_comment_message_tr_translated
> O Produto é bom, mas não deveria ter espaços tão grandes nos buracos,ouriças podem atravesar super recomendo muito bom	> The product is good, but it shouldn't have such large spaces in the holes, sea urchins can go through them, although, with difficulty. A... I highly recommend very good
> so tem demorado mais que o normal as entregas do site depois que começaram a vender com Loja horrível, não entregou o meu produto até hoje.	> It has only taken longer than usual to start selling with partners, other than that and more to talk about, it was delivered on time, the pr... Horrible store, didn't deliver my product until today.
> Mudo minha opinião assim que receber o produto. Consta no site que o produto foi entregue, mas O frete foi muito caro. Não valeu a pena. Porque eu tive que retirar a mercadoria na agência do consumidor. Cumpru e respeitou o consumidor.	> I change my opinion as soon as I receive the product. It says on the website that the product was delivered, but I didn't receive it. Shipping was very expensive. It wasn't worth it. Because I had to pick up the goods at the post office. Fulfilled and respected the consumer.
Não sei se recomendo. O produto foi marcado no site para um tipo de celular mas quando chegou AMEI O GORRO, É LINDO E COM ÓTIMO PREÇO. RECOMENDO SEM DÚVIDA!	> I don't know if I recommend it. The product was marked on the website for a type of cell phone, but when it arrived it did not fit on th... I LOVED THE BEANIE, IT'S BEAUTIFUL AND AT A GREAT PRICE. I RECOMMEND IT WITHOUT A DOUBT!
Pontualidade e ótimos produtos.	Punctuality and great products.
Tudo de acordo com o combinado.	Everything according to the agreement.
Os produtos vendidos através do stark são muito bons e sempre de qualidade.	The products sold through stark are very good and always of quality.
> Recebi o produto certo e antes do prazo. Valeu stark, targaryen/relojoaria nishimoto e correios. Gostei muito do produto, excelente	> I received the right product and ahead of schedule. Thanks stark, targaryen/nishimoto watchmaking and post offices. I really liked the product, excellent

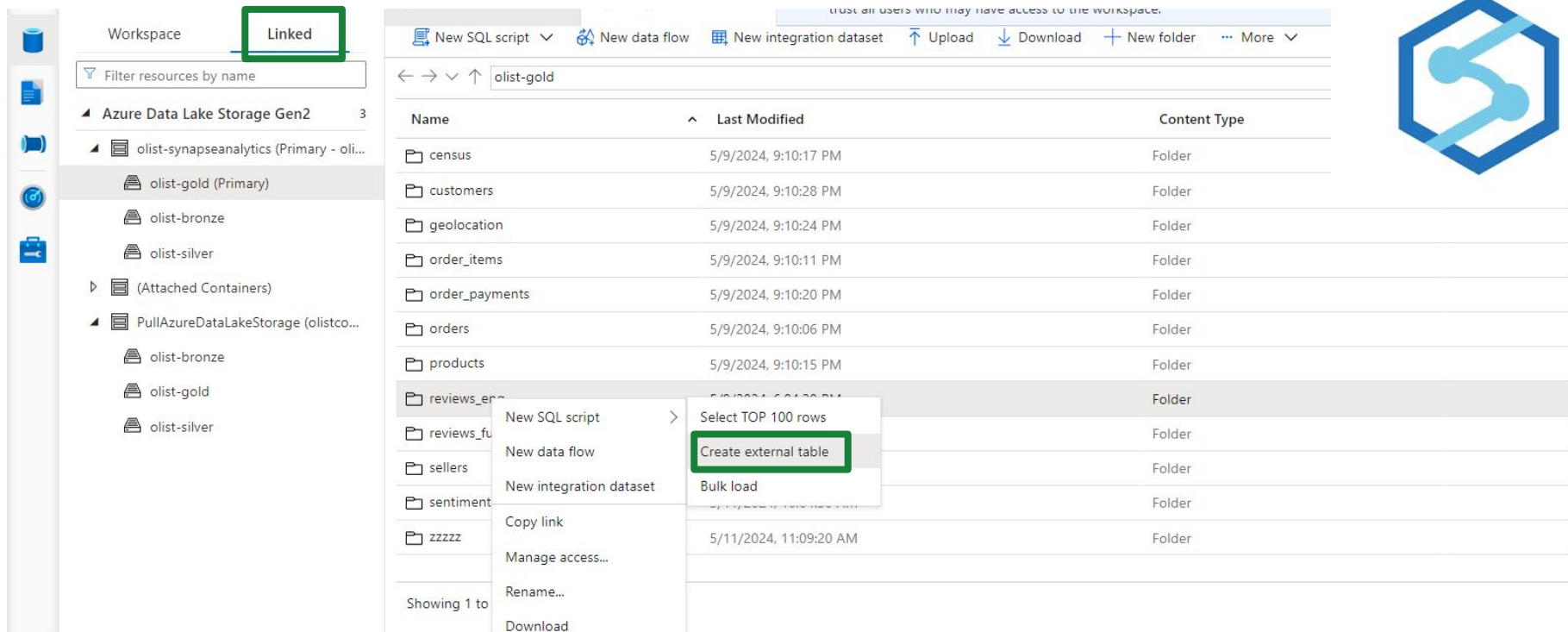


# Data Factory Progress

✓ Validate   ▶ Debug   ⚡ Add trigger



# Azure Synapse for Sentiment Analysis and Create SQL Queries – Loading



The screenshot displays the Azure Synapse Studio interface. On the left, the 'Workspace' pane shows a 'Linked' workspace. Below it, the 'Filter resources by name' search bar is visible. The 'Azure Data Lake Storage Gen2' container is expanded, showing a list of folders: 'olist-synapseanalytics (Primary - oli...', 'olist-gold (Primary)', 'olist-bronze', 'olist-silver', '(Attached Containers)', and 'PullAzureDataLakeStorage (olistco...'. The 'olist-gold (Primary)' folder is selected.

The main pane shows the 'olist-gold' container. A table lists the contents of the container, including folders like 'census', 'customers', 'geolocation', 'order\_items', 'order\_payments', 'orders', 'products', 'reviews\_en', 'reviews\_fu', 'sellers', 'sentiment', and 'zzzzz'. The 'reviews\_en' folder is selected, and a context menu is open over it. The menu options are: 'New SQL script', 'New data flow', 'New integration dataset', 'Copy link', 'Manage access...', 'Rename...', 'Download', 'Select TOP 100 rows', 'Create external table', and 'Bulk load'. The 'Create external table' option is highlighted.

The Azure Synapse logo is visible in the top right corner.

Name	Last Modified	Content Type
census	5/9/2024, 9:10:17 PM	Folder
customers	5/9/2024, 9:10:28 PM	Folder
geolocation	5/9/2024, 9:10:24 PM	Folder
order_items	5/9/2024, 9:10:11 PM	Folder
order_payments	5/9/2024, 9:10:20 PM	Folder
orders	5/9/2024, 9:10:06 PM	Folder
products	5/9/2024, 9:10:15 PM	Folder
reviews_en	5/9/2024, 9:10:20 PM	Folder
reviews_fu	5/9/2024, 9:10:20 PM	Folder
sellers	5/9/2024, 9:10:20 PM	Folder
sentiment	5/9/2024, 9:10:20 PM	Folder
zzzzz	5/11/2024, 11:09:20 AM	Folder

# Azure Synapse for Sentiment Analysis (Machine Learning Made Easy)

**Data** + ≡ <<

Workspace Linked

Filter resources by name

Lake database 2

- olist\_database
  - Tables
    - census
    - customers
    - geolocations
    - order\_items
    - order\_payments
    - orders
    - products
    - reviews\_eng
    - reviews\_full
    - sellers
  - olist\_new
- SQL database
  - olist\_gold (SQL)

Joins olist-gold x i

New SQL script New data flow New integration

olist-gold

Name	Last Modified
census	5/9/2024, 9:10:17 P
custom	5/9/2024, 9:10:28 P
geoloc	5/9/2024, 9:10:24 P
order_	5/9/2024, 9:10:11 P
order_	5/9/2024, 9:10:20 P
orders	5/9/2024, 9:10:06 P
produc	5/9/2024, 9:10:15 P
review:	5/9/2024, 6:04:20 P
review:	5/9/2024, 9:10:31 P
sellers	5/9/2024, 9:10:26 P
sentiment	5/11/2024, 10:04:50
	5/11/2024, 11:09:20

New SQL script  
New data flow  
New integration dataset  
Copy link  
Manage access...  
Rename...  
Download  
Delete  
Properties...

New SQL script  
New notebook  
Machine Learning

Train a new model  
Predict with a model

## Predict with a model

reviews\_full

## Choose a pre-trained model

### Azure Cognitive Services

This experience allows you to enrich the selected dataset with models.



#### Anomaly Detector

Anomaly detection is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data. [Learn more](#)



#### Sentiment Analysis

Evaluates the sentiment (positive/negative/neutral) of a text and also returns the probability (score) of the sentiment. [Learn more](#)

Continue

Cancel

# Azure Synapse for Sentiment Analysis

	review_comment_message_tr_translated	sentiment
	The company delivers within what it promises, on the last day of the deadline, it takes too long, it does not delight the customer, as it only does the basic obligation.	negative
45e	The product was delivered much earlier than expected. Great agility.	positive
066	Satisfied with the result.	positive
095	ON-TIME DELIVERY.. RECOMMEND!	positive
14	Arrived ahead of schedule and in perfect condition	positive
00	Great product, good prices and fast delivery.	positive
lc38l	loved the product	positive



# Azure Synapse for SQL Queries (Serverless SQL database)



Microsoft Azure | Synapse Analytics ▶ olist-synapseanalytics

Synapse live ▼ Validate all Publish all

Data + ≡ ≪

Workspace Linked

Filter resources by name

- ▶ Lake database 2
- ▶ SQL database 1
  - ▶ olist\_gold (SQL) ...
  - ▶ External tables ...
  - ▶ External resources
  - ▶ Views
  - ▶ Schemas
  - ▶ Security



- ▶ gold.products
- ▶ gold.review\_sentiments
- ▶ gold.reviews\_eng
- ▶ gold.reviews\_full
- ▶ gold.sellers

New SQL script ▶

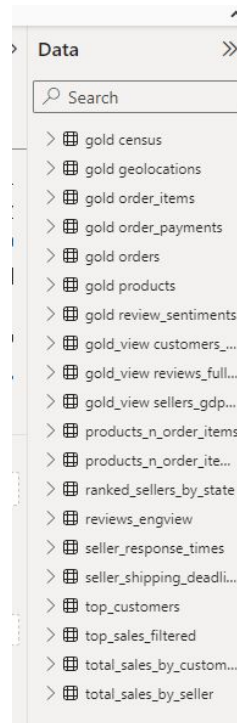
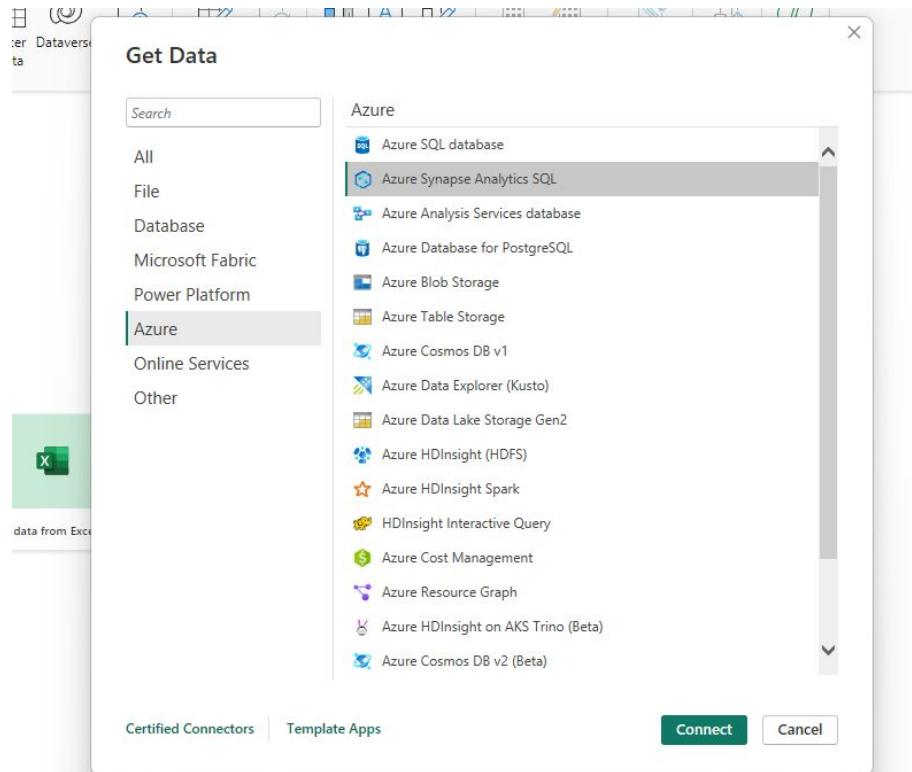
Refresh

# SQL Join Creation



```
1  -- Create the view gold.reviews_full_joined
2  CREATE VIEW gold.reviews_full_joined AS
3  SELECT
4      r.*,
5      oi.product_id,
6      oi.seller_id
7  FROM
8      gold.reviews_eng AS r
9  JOIN
10     (
11         SELECT DISTINCT
12             oi.product_id,
13             oi.seller_id,
14             oi.order_id
15         FROM
16             gold.order_item AS oi
17     ) AS oi
18 ON
19     r.order_id = oi.order_id;
```

# Loading Tables and View to Power BI



**Collaboration on  
Power Bi Service at  
[powerbi.com](https://powerbi.com)**

# Power BI

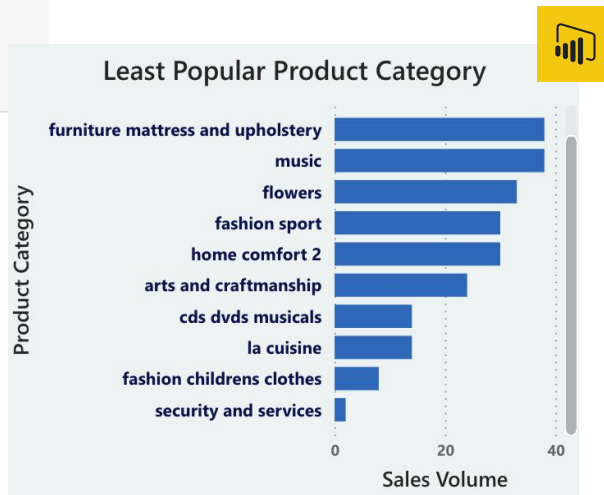
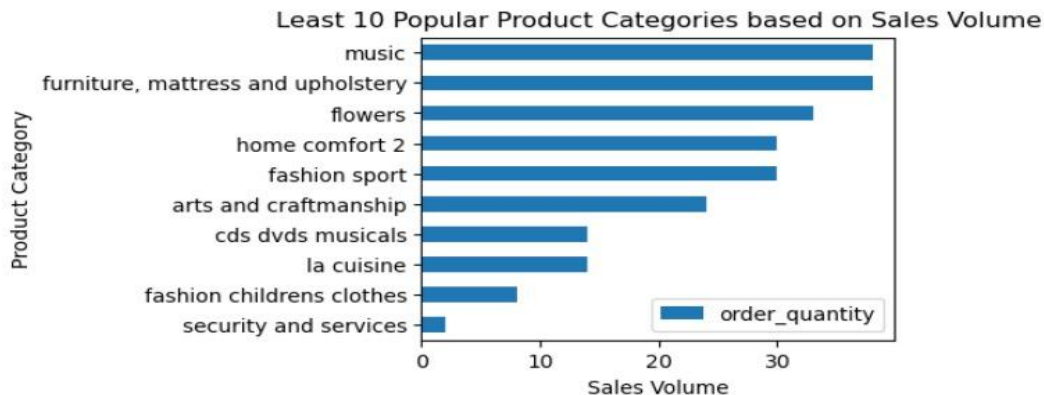
- Brazil's leading e-commerce marketplace, connecting small businesses to customers nationwide
- **BI goals:**
  - Increase sales turnover
  - Investigation of:
    - Sellers
    - Products
    - Customers





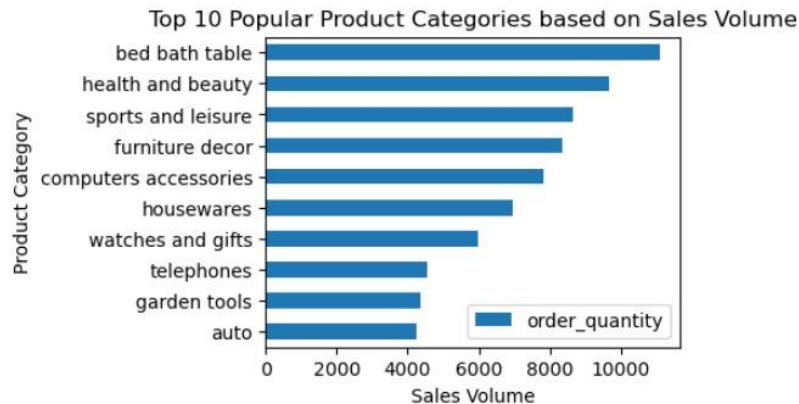
# Functional Testing (1)

```
2 # Select top 10 categories
3 least10_product_category = product_least_sales.head(10)
4
5 # Set the axis
6 fig, ax = plt.subplots(figsize=(4, 3))
7
8 # Plot the barplot with custom x and y axis
9 least10_product_category.sort_values(by='order_quantity').plot(x='product_category_name_english',
10                                                                y='order_quantity', ax=ax, kind='barh')
11
12 plt.xlabel('Sales Volume')
13 plt.ylabel('Product Category')
14 plt.title("Least 10 Popular Product Categories based on Sales Volume")
15 plt.show()
```

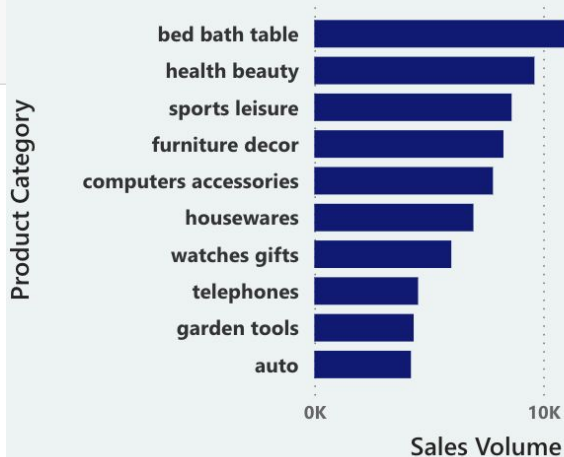


# Functional Testing (2)

```
3 # Select top 10 categories
4 top10_product_category = category_sales.head(10)
5
6 # Set the axis
7 fig, ax = plt.subplots(figsize=(4, 3))
8
9 # Plot the barplot with custom x and y axis
10 top10_product_category.sort_values(by='order_quantity').plot(x='product_category_name_english',
11                                                             y='order_quantity', ax=ax, kind='barh')
12
13 plt.xlabel('Sales Volume')
14 plt.ylabel('Product Category')
15 plt.title("Top 10 Popular Product Categories based on Sales Volume")
16 plt.show()
17
```



Top 10 Product Category



Thank You

Q&A

The image features a dark blue background on the left side, which transitions into a white background on the right. A thin, gold-colored diagonal line runs from the bottom left towards the center. The text 'Q&A' is written in a white, serif font, positioned in the upper left area of the dark blue section.