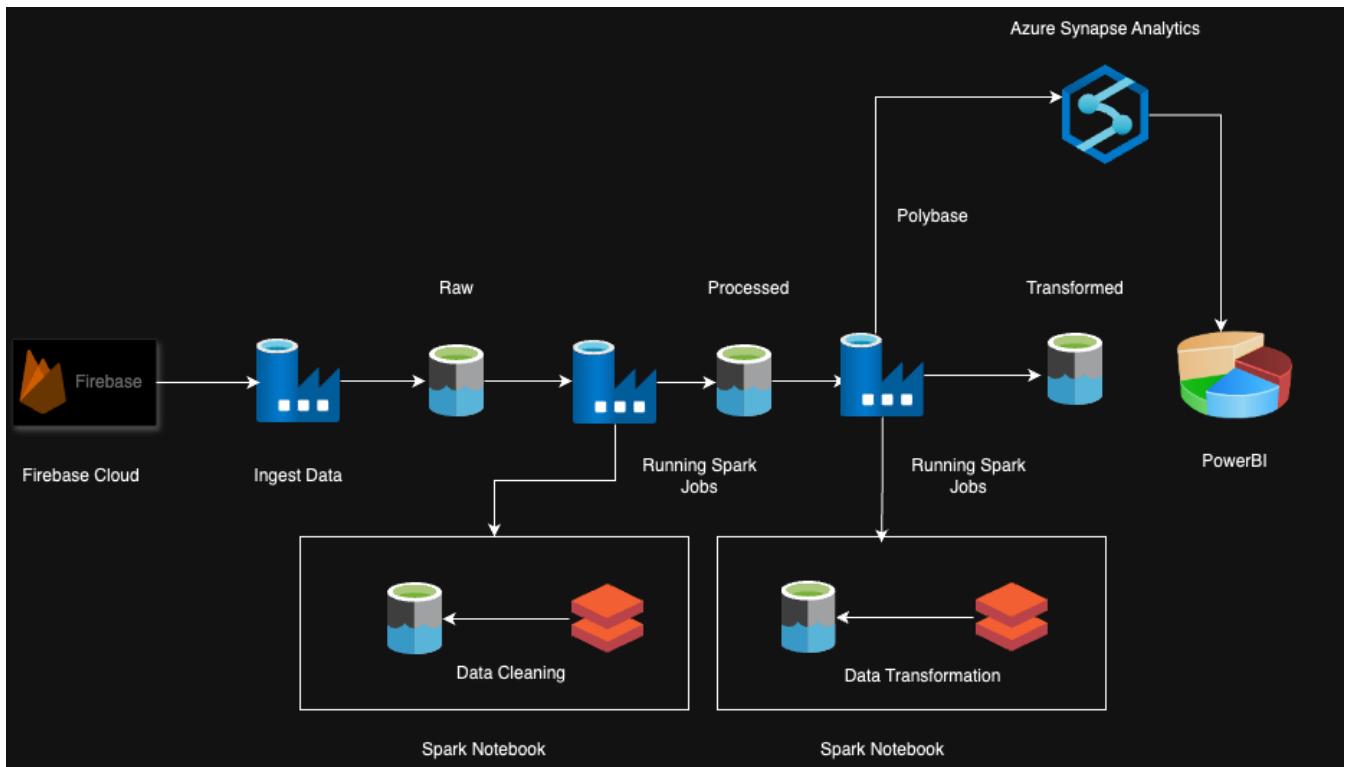


Shreehari

Meet Patel

## PROJECT – RETAIL SALES ANALYSIS

## Architecture Diagram



## Raw Data of Online Retail in Excel Format

Microsoft Excel File Edit View Insert Format Tools Data Window Help

AutoSave Online Retail

Home Insert Draw Page Layout Formulas Data Review View Automate Acrobat Tell me

Cut Copy Paste Format Calibri (Body) 11 A A Wrap Text v Merge & Center General Conditional Formatting Format as Table Normal Bad Good Neutral AutoSum Fill Sort & Filter Insert Delete Format Comments Share

N24

Open recovered workbooks? Your recent changes were saved. Do you want to continue working where you left off?

| InvoiceNo | StockCode | Description                   | Quantity | InvoiceDate     | UnitPrice | CustomerID | Country        |
|-----------|-----------|-------------------------------|----------|-----------------|-----------|------------|----------------|
| 2         | S36365    | 85123A WHITE HANGING HEAI     | 6        | 2010-12-01 8:26 | 2.55      | 17850      | United Kingdom |
| 3         | S36365    | 71053 WHITE METAL LANTER      | 6        | 2010-12-01 8:26 | 3.39      | 17850      | United Kingdom |
| 4         | S36365    | 84406B CREAM/CUPID HEARTS     | 8        | 2010-12-01 8:26 | 2.75      | 17850      | United Kingdom |
| 5         | S36365    | 84293A CREAM/CUPID HEARTS     | 6        | 2010-12-01 8:26 | 2.75      | 17850      | United Kingdom |
| 6         | S36365    | 84028E RED WOOLLY HOTTIE!     | 6        | 2010-12-01 8:26 | 3.39      | 17850      | United Kingdom |
| 7         | S36365    | 22752 SETT BABUSHKA NEST      | 2        | 2010-12-01 8:26 | 7.65      | 17850      | United Kingdom |
| 8         | S36365    | 21730 GLASS STAR FROSTED      | 6        | 2010-12-01 8:26 | 4.25      | 17850      | United Kingdom |
| 9         | S36366    | 22633 HAND WARMER UNIO        | 6        | 2010-12-01 8:28 | 1.85      | 17850      | United Kingdom |
| 10        | S36366    | 22633 HAND WARMER UNIO        | 6        | 2010-12-01 8:28 | 1.85      | 17850      | United Kingdom |
| 11        | S36367    | 84879 CREAM/COLOUR BI         | 32       | 2010-12-01 8:34 | 1.69      | 13047      | United Kingdom |
| 12        | S36367    | 22745 POPPY'S PLAYHOUSE E     | 6        | 2010-12-01 8:34 | 2.1       | 13047      | United Kingdom |
| 13        | S36367    | 22748 POPPY'S PLAYHOUSE K     | 6        | 2010-12-01 8:34 | 2.1       | 13047      | United Kingdom |
| 14        | S36367    | 22749 FELTCRAFT PRINCESS C    | 8        | 2010-12-01 8:34 | 3.75      | 13047      | United Kingdom |
| 15        | S36367    | 22310 IVORY KNITTED MUG C     | 6        | 2010-12-01 8:34 | 1.65      | 13047      | United Kingdom |
| 16        | S36367    | 84961 BOX OF ASSORTED C       | 6        | 2010-12-01 8:34 | 4.25      | 13047      | United Kingdom |
| 17        | S36368    | 22623 BOX OF VINTAGE ALPH     | 3        | 2010-12-01 8:34 | 4.25      | 13047      | United Kingdom |
| 18        | S36368    | 22622 BOX OF VINTAGE ALPH     | 2        | 2010-12-01 8:34 | 9.95      | 13047      | United Kingdom |
| 19        | S36367    | 21754 HOME BUILDING BLOC      | 3        | 2010-12-01 8:34 | 5.95      | 13047      | United Kingdom |
| 20        | S36367    | 21755 LOVE BUILDING BLOC      | 3        | 2010-12-01 8:34 | 5.95      | 13047      | United Kingdom |
| 21        | S36367    | 21777 RECIP BOX WITH MET      | 4        | 2010-12-01 8:34 | 7.95      | 13047      | United Kingdom |
| 22        | S36367    | 48187 DOORMAT NEW ENGL        | 4        | 2010-12-01 8:34 | 7.95      | 13047      | United Kingdom |
| 23        | S36368    | 22962 JAM MAKING SET/THI      | 6        | 2010-12-01 8:34 | 4.25      | 13047      | United Kingdom |
| 24        | S36368    | 22911 JAM MAKING SET/THI      | 3        | 2010-12-01 8:34 | 4.25      | 13047      | United Kingdom |
| 25        | S36368    | 22912 YELLOW COAT RACK P.     | 3        | 2010-12-01 8:34 | 4.95      | 13047      | United Kingdom |
| 26        | S36368    | 22914 BLUE COAT RACK PARI     | 3        | 2010-12-01 8:34 | 4.95      | 13047      | United Kingdom |
| 27        | S36369    | 21756 BATH ALARM CLOCK BAKELI | 3        | 2010-12-01 8:35 | 5.95      | 13047      | United Kingdom |
| 28        | S36370    | 22722 ALARM CLOCK BAKELI      | 24       | 2010-12-01 8:45 | 3.75      | 12583      | France         |
| 29        | S36370    | 22723 ALARM CLOCK BAKELI      | 24       | 2010-12-01 8:45 | 3.75      | 12583      | France         |
| 30        | S36370    | 22726 ALARM CLOCK BAKELI      | 24       | 2010-12-01 8:45 | 3.75      | 12583      | France         |
| 31        | S36370    | 21724 PANDA AND RUNNES        | 12       | 2010-12-01 8:45 | 0.85      | 12583      | France         |
| 32        | S36370    | 21883 STARS GIFT TAPE         | 24       | 2010-12-01 8:45 | 0.65      | 12583      | France         |
| 33        | S36370    | 10002 INFATTABLE POLITICA     | 48       | 2010-12-01 8:45 | 0.85      | 12583      | France         |
| 34        | S36370    | 21793 VINTAGE HEADS AND 1     | 24       | 2010-12-01 8:45 | 1.25      | 12583      | France         |
| 35        | S36370    | 21035 SETZ RED RETROSPOT      | 18       | 2010-12-01 8:45 | 2.95      | 12583      | France         |
| 36        | S36370    | 22320 VINTAGE HEADS AND 1     | 24       | 2010-12-01 8:45 | 2.95      | 12583      | France         |
| 37        | S36370    | 22629 9PC ECRY LUNCH BOX      | 24       | 2010-12-01 8:45 | 1.85      | 12583      | France         |
| 38        | S36370    | 22659 LUNCH BOX I LOVE LO     | 24       | 2010-12-01 8:45 | 1.95      | 12583      | France         |
| 39        | S36370    | 22631 CIRCUS PARADE LUNC      | 24       | 2010-12-01 8:45 | 1.95      | 12583      | France         |
| 40        | S36370    | 22661 CHARLOTTE BAG DOLL      | 20       | 2010-12-01 8:45 | 0.85      | 12583      | France         |
| 41        | S36370    | 21731 RED TOADSTOOL LED F     | 24       | 2010-12-01 8:45 | 1.65      | 12583      | France         |
| 42        | S36370    | 22902 SET 2 TEA TOWELS LIC    | 24       | 2010-12-01 8:45 | 2.95      | 12583      | France         |
| 43        | S36370    | 22311 VINTAGE HEADS JIGSAW    | 24       | 2010-12-01 8:45 | 2.95      | 12583      | France         |
| 44        | S36370    | 22540 MINI JIGSAW CIRCUS I    | 24       | 2010-12-01 8:45 | 0.42      | 12583      | France         |
| 45        | S36370    | 22544 MINI JIGSAW SPACEBK     | 24       | 2010-12-01 8:45 | 0.42      | 12583      | France         |
| 46        | S36370    | 22492 MINI PAINT SET VINTA    | 36       | 2010-12-01 8:45 | 0.65      | 12583      | France         |
| 47        | S36370    | POST POSTAGE                  | 3        | 2010-12-01 8:45 | 1.28      | 12583      | France         |
| 48        | S36371    | 20905 MINI PAINT KIT 50'S     | 80       | 2010-12-01 9:00 | 2.55      | 17850      | United Kingdom |
| 49        | S36372    | 22652 HAND WARMER RED P       | 6        | 2010-12-01 9:01 | 1.85      | 17850      | United Kingdom |
| 50        | S36372    | 22623 HAND WARMER UNIO        | 6        | 2010-12-01 9:01 | 1.85      | 17850      | United Kingdom |
| 51        | S36373    | 85123A WHITE HANGING HEAI     | 6        | 2010-12-01 9:02 | 2.55      | 17850      | United Kingdom |
| 52        | S36373    | 71053 WHITE METAL LANTER      | 6        | 2010-12-01 9:02 | 3.39      | 17850      | United Kingdom |

# Azure Data Factory and Azure Data Lake Storage Account

## Creating Azure Data Lake Storage Account and Azure Data Factory

The screenshot shows the Microsoft Azure portal interface. At the top, there's a navigation bar with 'Microsoft Azure', an 'Upgrade' button, a search bar ('Search resources, services, and docs (G+/)'), and user information ('Shriji74@outlook.com DEFAULT DIRECTORY'). Below the navigation bar, the main content area shows a deployment named 'retailsadls\_1717367632299' with a status message: 'Your deployment is complete'. It provides deployment details: Deployment name: 'retailsadls\_1717367632299', Subscription: 'Shriji-Wp', Resource group: 'retail-rg'. It also shows the start time: 02/06/2024, 18:34:04 and Correlation ID: 12516d3a-dc6c-4851-af6a-481ac4b29026. There are buttons for 'Delete', 'Cancel', 'Redeploy', 'Download', and 'Refresh'. On the left, there's a sidebar with navigation links: Home, Overview (selected), Inputs, Outputs, and Template. Below the main content, there are several promotional cards: 'Cost Management' (Get notified to stay within your budget and prevent unexpected charges on your bill. Set up cost alerts >), 'Microsoft Defender for Cloud' (Secure your apps and infrastructure. Go to Microsoft Defender for Cloud >), 'Free Microsoft tutorials' (Start learning today >), and 'Work with an expert' (Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support. Find an Azure expert >). At the bottom left, there's a URL: 'https://portal.azure.com/#'

Microsoft Azure Upgrade Search resources, services, and docs (G+/)

Home > Microsoft.DataFactory-20240602183606 | Overview Deployment

Search Delete Cancel Redeploy Download Refresh

Deployment succeeded Deployment 'Microsoft.DataFactory-20240602183606' to resource group 'retail-rg' was successful.

Pin to dashboard Go to resource gr...

Overview Inputs Outputs Template

Your deployment is complete

Deployment name : Microsoft.DataFactory-20240602183606 Start time : 02/06/2024, 18:36:28  
Subscription : Shriji-Wp Correlation ID : ed9355c5-43a9-434b-82e4-b9b18a491d5f  
Resource group : retail-rg

Deployment details Next steps Go to resource

Give feedback Tell us about your experience with deployment

Cost management Get notified to stay within your budget and prevent unexpected charges on your bill. Set up cost alerts >

Microsoft Defender for Cloud Secure your apps and infrastructure Go to Microsoft Defender for Cloud >

Free Microsoft tutorials Start learning today >

Work with an expert Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support. Find an Azure expert >

## Creating different Directory in ADLS Container based on Medallion Architecture.

The screenshot shows the Microsoft Azure Storage Explorer interface for a storage account named 'retailsadls'. The left sidebar navigation includes 'Overview', 'Activity log', 'Tags', 'Diagnose and solve problems', 'Access Control (IAM)', 'Data migration', 'Events', 'Storage browser', and 'Data storage' (with 'Containers' selected). The main content area displays a table of containers:

| Name           | Last modified        | Anonymous access level | Lease state |
|----------------|----------------------|------------------------|-------------|
| \$logs         | 02/06/2024, 18:34:27 | Private                | Available   |
| retail-project | 02/06/2024, 18:35:02 | Private                | Available   |

A success message in the top right corner states: 'Successfully created storage container 'retail-project''. The top navigation bar shows the user's email 'Shriji74@outlook.com' and the 'DEFAULT DIRECTORY' setting.

Microsoft Azure Upgrade Search resources, services, and docs (G+)

Home > retailadls\_1717367632299 | Overview > retailadls | Containers >

**retail-project** Container

Successfully added directory  
Successfully added directory 'transformation'.

Search  Upload Add Directory Refresh Rename Delete Change tier Acquire lease Break lease Give feedback

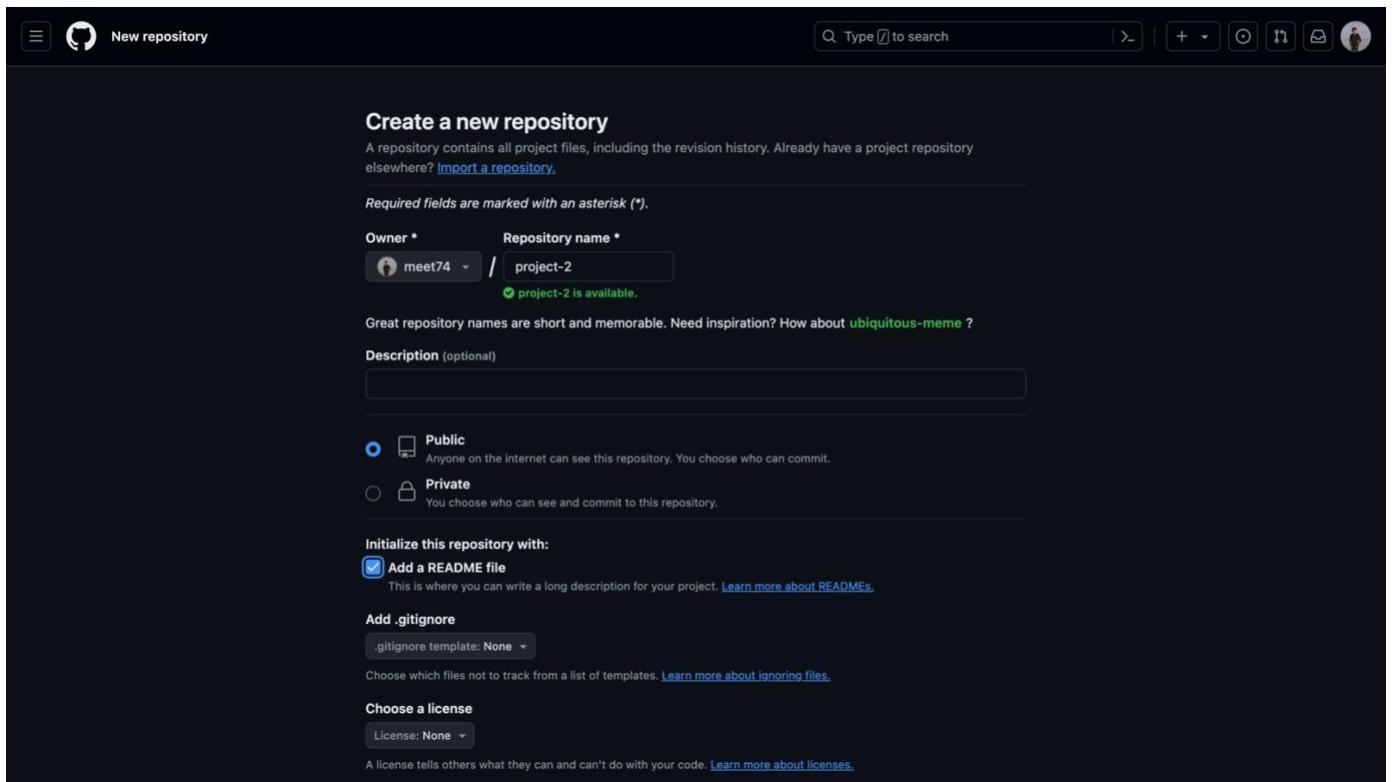
Overview Diagnose and solve problems Access Control (IAM) Settings

Authentication method: Access key (Switch to Microsoft Entra user account)  
Location: retail-project

Search blobs by prefix (case-sensitive) Show deleted objects

| Name           | Modified | Access tier | Archive status | Blob type | Size | Lease state |
|----------------|----------|-------------|----------------|-----------|------|-------------|
| processed      |          |             |                |           | -    | ***         |
| raw            |          |             |                |           | -    | ***         |
| transformation |          |             |                |           | -    | ***         |

## Creating a Git Repository



Used Data Factory to fetch data from Firebase cloud through http linked service and stored in Azure Data Lake Storage Account

## Git Integration in Azure Data Factory

The screenshot shows the 'Git repository' configuration page in the Azure Data Factory portal. The left sidebar lists various settings like Home, Author, Monitor, Manage, and Learning Center. The main panel shows the 'Git repository' configuration with the following details:

| Setting                   | Value                                   |
|---------------------------|---|
| Repository type           | GitHub                                  |
| GitHub account            | meet74                                  |
| Repository name           | project-2                               |
| Collaboration branch      | main                                    |
| Publish branch            | adf_publish                             |
| Root folder               | /                                       |
| Last published commit     | 5b8f1893ca63fc71d7a634236b8a51d9922977c |
| Publish (from ADF Studio) | Enabled                                 |
| Custom comment            | Enabled                                 |

## Creating HTTP and ADLS Linked Service and Dataset

The screenshot shows the Microsoft Azure Data Factory interface for the factory 'retailsadf'. The left sidebar navigation menu includes General, Connections, Linked services (which is selected), Integration runtimes, Microsoft Purview, Source control, Author, Security, Workflow orchestration manager, and Apache Airflow. The main content area is titled 'Linked services' and contains a sub-section 'New linked service'. A dropdown menu for 'HTTP' is open. The configuration fields include:

- Name \***: ls\_http\_retail\_input
- Description**: (empty)
- Connect via integration runtime \***: AutoResolveIntegrationRuntime
- Base URL \***: https://firebasestorage.googleapis.com
- Information**: Information will be sent to the URL specified. Please ensure you trust the URL entered.
- Server certificate validation**: Disable (radio button selected)
- Authentication type \***: Anonymous
- Auth headers**: New
- Annotations**: New
- Parameters**: Advanced

At the bottom right, there is a message 'Connection successful' with a checkmark icon, and buttons for 'Creating...', 'Back', 'Test connection', and 'Cancel'.

The screenshot shows the Microsoft Azure Data Factory interface. On the left, a sidebar lists various settings: General, Factory settings, Connections (Linked services selected), Integration runtimes, Microsoft Purview, Source control (Git configuration, ARM template), Author (Triggers, Global parameters, Data flow libraries), Security (Credentials, Customer managed key, Outbound rules, Managed private endpoints), Workflow orchestration manager, and Apache Airflow.

The main area displays a "Linked services" section. A table shows one item: "ls\_http\_retail\_input" (Type: HTTP). To the right, a "New linked service" form is open:

- Name:** ls\_retail\_raw\_output
- Connect via integration runtime:** AutoResolveIntegrationRuntime
- Authentication type:** Account key
- Account selection method:** From Azure subscription (selected)
- Azure subscription:** Shriji-Wp (9b093c33-a116-4e12-aa43-073a1c622e60)
- Storage account name:** retailads1
- Test connection:** To linked service (selected)
- Annotations:** + New

At the bottom right of the form, there are "Create", "Back", "Test connection" (with a green checkmark), and "Cancel" buttons.

The screenshot shows the Microsoft Azure Data Factory interface. The left sidebar shows Factory Resources: Pipelines (0), Change Data Capture (preview) (0), Datasets (2), Data flows (0), Power Query (0), and Templates (0).

The main area shows two datasets: "ds\_retail\_adls\_output" (DelimitedText) and "ds\_retail\_http\_input" (CSV). The "ds\_retail\_http\_input" dataset is selected. Its properties are displayed on the right:

- Name:** ds\_retail\_http\_input
- Description:** (empty)
- Annotations:** + New

The "Connection" tab of the dataset configuration is shown, with the following details:

- Linked service:** ls\_http\_retail\_input
- Base URL:** https://firebasestorage.googleapis.com/
- Relative URL:** /v0/b/music-player-98871.appspot.com/...
- Compression type:** Select...
- Column delimiter:** Comma (,)
- Row delimiter:** Default (\r\n, or \n\r)
- Encoding:** Default(UTF-8)
- Quote character:** Double quote (")
- Escape character:** Backslash (\)
- First row as header:** checked

## Creating Pipeline

The screenshot shows the Microsoft Azure Data Factory interface for creating a pipeline. On the left, the 'Factory Resources' sidebar lists 'Pipelines' (1), 'Datasets' (2), and other resources. In the center, a pipeline named 'pl\_http\_fetch\_retail\_data' is being edited. The 'Activities' pane shows a 'Copy data' activity named 'copy\_http\_data'. The 'Properties' pane on the right displays the activity's configuration, including:

- Name:** pl\_http\_fetch\_retail\_data
- Description:** (empty)
- Activity state:** Activated (radio button selected)
- Timeout:** 0.12:00:00
- Retry:** 0
- Retry interval (sec):** 30

Pipeline Succeeded

The screenshot shows the Microsoft Azure Data Factory interface displaying the status of a pipeline run. The pipeline 'pl\_http\_fetch\_retail\_data' is shown as 'Succeeded'. The 'Output' section provides details of the run:

- Pipeline run ID:** Sa3e7272-ecac-47de-9745-8f5f9fa0a410
- Pipeline status:** Succeeded
- Run start:** 6/2/2024, 7:32:14 PM
- Duration:** 14s
- Integration runtime:** AutoResolveIntegratio

| Activity name  | Activity status | Activity type | Run start            | Duration | Integration runtime   | User properties |
|----------------|-----------------|---------------|----------------------|----------|-----------------------|-----------------|
| copy_http_data | Succeeded       | Copy data     | 6/2/2024, 7:32:14 PM | 14s      | AutoResolveIntegratio |                 |

Transformed CSV file in ADLS.

The screenshot shows the Microsoft Azure Storage Explorer interface. At the top, there's a navigation bar with 'Microsoft Azure' and 'Upgrade' buttons, a search bar, and user information ('Shriji74@outlook.com DEFAULT DIRECTORY'). Below the navigation bar, the path 'Home > retailads | Containers > retail-project' is displayed. The 'Container' section for 'retail-project' is shown, with tabs for 'Overview' (which is selected), 'Diagnose and solve problems', 'Access Control (IAM)', and 'Settings'. The 'Overview' tab displays the location 'retail-project / raw' and a search bar for blobs by prefix. A table lists a single blob named 'online-retail'. The table columns are: Name, Modified, Access tier, Archive status, Blob type, Size, and Lease state. The blob details are: Name - online-retail, Modified - 02/06/2024, 19:33:43, Access tier - Hot (Inferred), Archive status - Not yet archived, Blob type - Block blob, Size - 43.95 MiB, Lease state - Available. There's also a 'Show deleted objects' toggle switch.

| Name          | Modified             | Access tier    | Archive status   | Blob type  | Size      | Lease state |
|---------------|----------------------|----------------|------------------|------------|-----------|-------------|
| online-retail | 02/06/2024, 19:33:43 | Hot (Inferred) | Not yet archived | Block blob | 43.95 MiB | Available   |

# Databricks

## Creating Databricks Workspace

The screenshot shows the Microsoft Azure Deployment Overview page for a deployment named "retail-rg\_retaildbrw". The deployment is marked as complete with a green checkmark icon. The deployment details are listed as follows:

| Deployment name | : retail-rg_retaildbrw |
|-----------------|------------------------|
| Subscription    | : Shriji-Wp            |
| Resource group  | : retail-rg            |

The deployment started at 02/06/2024, 19:37:50 with a Correlation ID of a7b07adb-8a3d-49b3-8af9-116ab2f6f76f. The page also includes sections for "Deployment details", "Next steps", and "Give feedback". On the right side, there are promotional cards for "Cost management", "Microsoft Defender for Cloud", and "Work with an expert".

## Mounting Databricks to Azure Data Lake Storage Account using Service Principle

App Registration in Microsoft Intra ID

The screenshot shows the 'Register an application' page in the Microsoft Azure portal. The top navigation bar includes 'Microsoft Azure', 'Upgrade', a search bar, and user information ('Shriji74@outlook.com'). The main content area has a title 'Register an application' with a back arrow. A required field 'Name' is filled with 'retail-app'. Below it, 'Supported account types' are listed with the first option ('Accounts in this organizational directory only') selected. A note says 'Who can use this application or access this API?'. The 'Redirect URI (optional)' section contains a dropdown for platform selection and a text input for the URI ('e.g. https://example.com/auth'). A note says 'We'll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.' At the bottom, there's a link 'By proceeding, you agree to the Microsoft Platform Policies' and a blue 'Register' button.

## Used Key Vault and created Secrets.

Microsoft Azure Upgrade Search resources, services, and docs (G+)

Home > retail-key-vault | Overview > retail-key-vault | Secrets >

**retail-client** Versions

New Version Refresh Delete Download Backup

| Version                          | Status    | Activation date | Expiration date |
|----------------------------------|-----------|-----------------|-----------------|
| CURRENT VERSION                  | ✓ Enabled |                 |                 |
| 1df9c7e948ab478fae967127b0fa244b |           |                 |                 |

Give feedback

Microsoft Azure Upgrade Search resources, services, and docs (G+)

Home > retail-key-vault | Overview > retail-key-vault | Secrets >

**Create a secret**

Upload options: Manual

Name \*: retail-tenant

Secret value \*: \*\*\*\*\*

Content type (optional):

Set activation date:

Set expiration date:

Enabled: Yes

Tags: 0 tags

Create Cancel

**retail-app | Certificates & secrets**

**Certificates (0) Client secrets (1) Federated credentials (0)**

| Description | Expires    | Value  | Secret ID                            |
|-------------|------------|--|--------------------------------------|
| retail      | 29/11/2024 | ByG8Q~DCctQeSzlp8KOIZXE9Wth..~Vb4t... [REDACTED] | 46bf14b3-ac90-4a30-b08f-31f542c4d144 |

**Create a secret**

**Upload options**

**Name \***

**Secret value \***

**Content type (optional)**

**Set activation date**

**Set expiration date**

**Enabled**  Yes  No

**Tags** 0 tags

**Create** **Cancel**

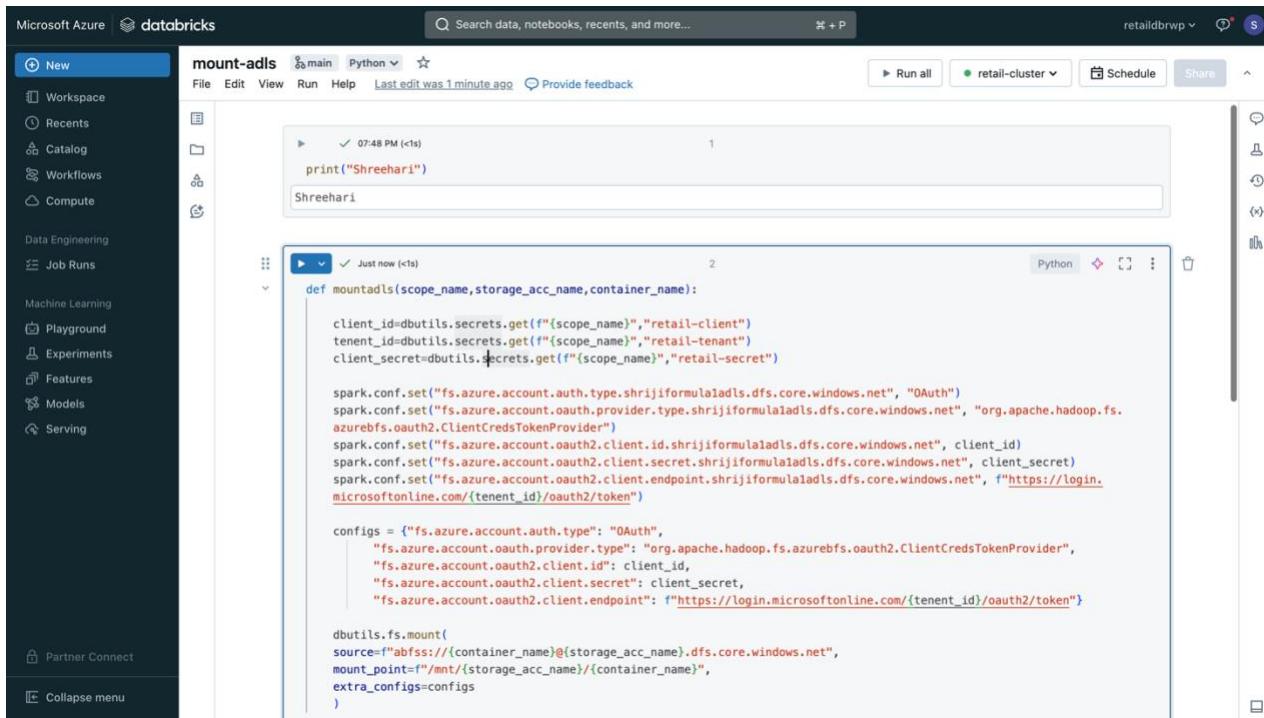
## Created Secret Scope in Databricks

The screenshot shows the 'Create Secret Scope' page in the Databricks UI. On the left, there's a sidebar with various workspace options like Workspace, Recents, Catalog, Workflows, Compute, Data Engineering, Job Runs, Machine Learning, Playground, Experiments, Features, Models, and Serving. The main area has a title 'Create Secret Scope' with 'Cancel' and 'Create' buttons. A descriptive text explains that it's a store for secrets identified by a name and backed by a specific store type, with a 'Learn more' link. The 'Scope Name' field contains 'retail-scope'. The 'Manage Principal' dropdown is set to 'All workspace users'. Under 'Azure Key Vault', the 'DNS Name' field is 'https://retail-key-vault.vault.azure.net/' and the 'Resource ID' field shows a partially typed URL: '30/resourceGroups/retail-rg/providers/Microsoft.KeyVault/vaults/retail-key-vault'.

## Given RBAC Permission to retail-app.

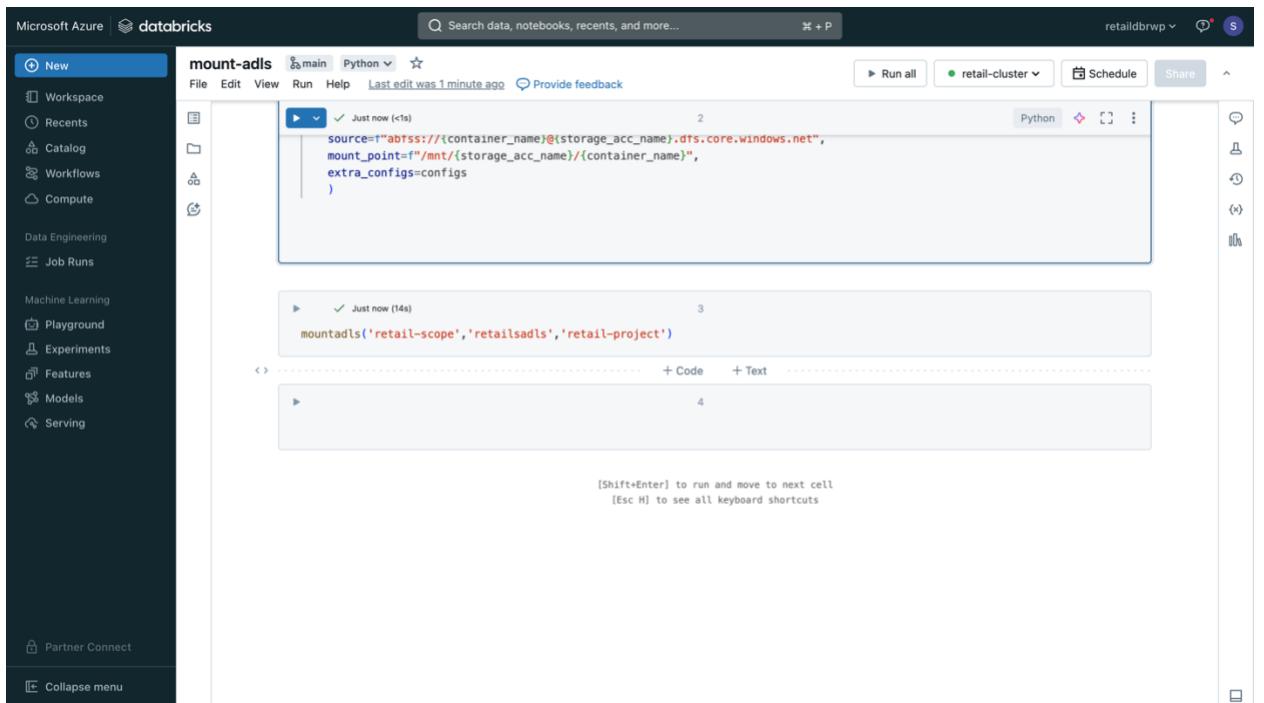
The screenshot shows the Microsoft Azure interface for managing role assignments. On the left, the 'Add role assignment' page is displayed with the 'Members' tab selected. It shows a 'Selected role' of 'Storage Blob Data Contributor' and 'Assign access to' set to 'User, group, or service principal'. The 'Members' section has a search bar with 'ret' and a result for 'retailsadf'. On the right, a 'Select members' dialog box is open, showing the selected member 'retail-app' with a 'Remove' link. Navigation buttons at the bottom include 'Review + assign', 'Previous', 'Next', 'Select', and 'Close'.

## Mounted ADLS to Databricks



The screenshot shows a Databricks notebook interface with the following details:

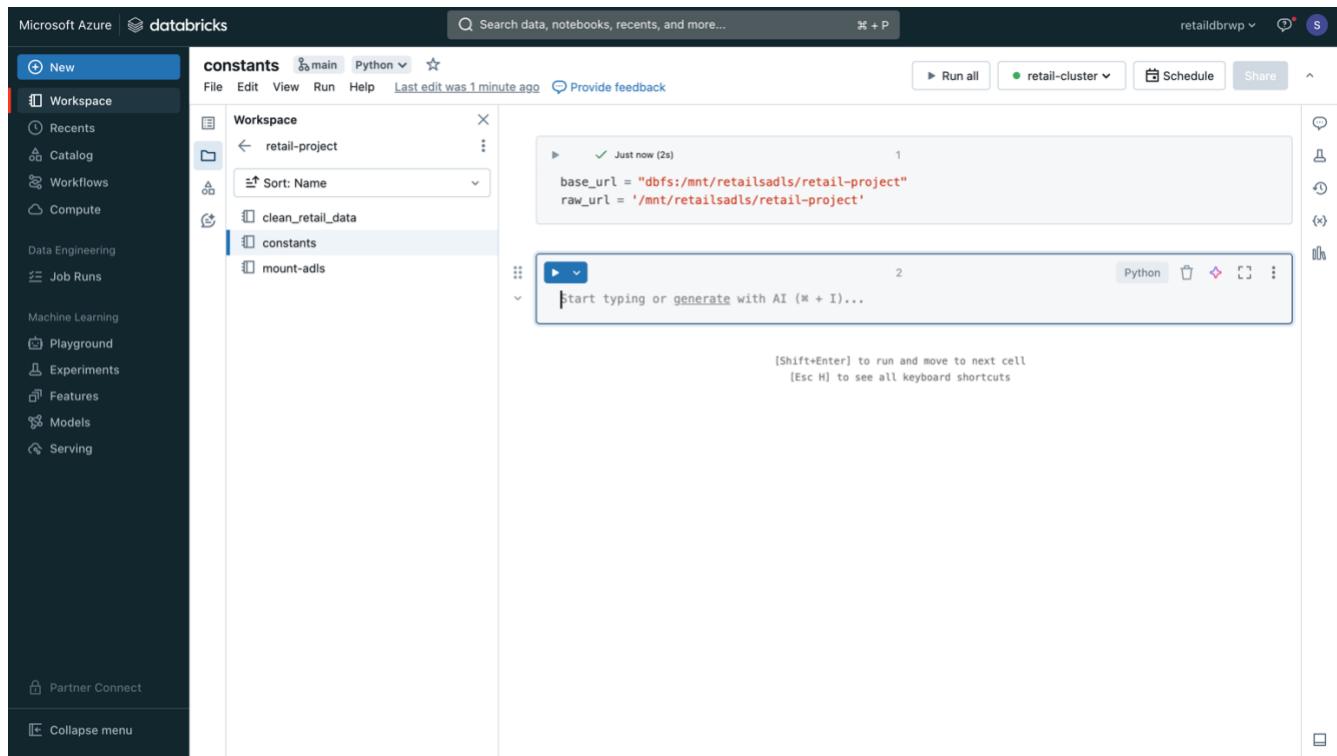
- Title:** mount-adls
- Languages:** Python
- Run Status:** Just now (1s)
- Code Cells:**
  - print("Shreehari")
  - def mountadls(scope\_name,storage\_acc\_name,container\_name):
 client\_id=dbutils.secrets.get(f"{scope\_name}","retail-client")
 tenant\_id=dbutils.secrets.get(f"{scope\_name}","retail-tenant")
 client\_secret=dbutils.secrets.get(f"{scope\_name}","retail-secret")
 
 spark.conf.set("fs.azure.account.auth.type", "OAuth")
 spark.conf.set("fs.azure.account.oauth.provider.type", "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider")
 spark.conf.set("fs.azure.account.oauth2.client.id", shrijiformulaadls.dfs.core.windows.net)
 spark.conf.set("fs.azure.account.oauth2.client.secret", shrijiformulaadls.dfs.core.windows.net)
 spark.conf.set("fs.azure.account.oauth2.client.endpoint", "https://login.microsoftonline.com/{tenant\_id}/oauth2/token")
 
 configs = {"fs.azure.account.auth.type": "OAuth",
 "fs.azure.account.oauth.provider.type": "org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider",
 "fs.azure.account.oauth2.client.id": client\_id,
 "fs.azure.account.oauth2.client.secret": client\_secret,
 "fs.azure.account.oauth2.client.endpoint": "https://login.microsoftonline.com/{tenant\_id}/oauth2/token"}
 
 dbutils.fs.mount(
 source=f"abfss://{container\_name}@{storage\_acc\_name}.dfs.core.windows.net",
 mount\_point=f"/mnt/{storage\_acc\_name}/{container\_name}",
 extra\_configs=configs
 )
- Output:** Shreehari



The screenshot shows a Databricks notebook interface with the following details:

- Title:** mount-adls
- Languages:** Python
- Run Status:** Just now (1s)
- Code Cells:**
  - source="abfss://{container\_name}@{storage\_acc\_name}.dfs.core.windows.net",
 mount\_point="/mnt/{storage\_acc\_name}/{container\_name}",
 extra\_configs=configs
 )
  - mountadls('retail-scope','retailsadls','retail-project')
  - 
  -
- Output:** [Shift+Enter] to run and move to next cell  
[Esc H] to see all keyboard shortcuts

## Constants File to store constant data



The screenshot shows the Microsoft Azure Databricks workspace interface. On the left, a sidebar lists various workspace sections: Recents, Catalog, Workflows, Compute, Data Engineering, Job Runs, Machine Learning, Playground, Experiments, Features, Models, and Serving. A 'Partner Connect' link and a 'Collapse menu' button are also present.

The main area displays a notebook titled 'constants'. The sidebar shows a tree structure for the 'retail-project' catalog, with branches for 'clean\_retail\_data', 'constants', and 'mount-adls'. The notebook itself has two cells:

```
base_url = "dbfs:/mnt/retailsadls/retail-project"  
raw_url = '/mnt/retailsadls/retail-project'
```

Cell 1: Shows the code above. Cell 2: A text input field with placeholder text: '\$start typing or generate with AI (⌘ + I)...'. Below the cells, keyboard shortcuts are listed: [Shift+Enter] to run and move to next cell, and [Esc H] to see all keyboard shortcuts.

Loaded Raw Data from ADLS

The screenshot shows the Databricks workspace interface. On the left, the sidebar includes options like New, Workspace, Recents, Catalog, Workflows, Compute, Data Engineering, Job Runs, Machine Learning, Playground, Experiments, Features, Models, and Serving. The main area displays a Python notebook titled "clean\_retail\_data". The code defines a schema for a CSV file and reads it into a DataFrame named "retail\_df". Below the code, a "display" command is used to show the DataFrame as a table. The table has columns: Quantity, InvoiceDate, UnitPrice, CustomerID, and Country.

```
schema = StructType([
    StructField("InvoiceNo", IntegerType(), True),
    StructField("StockCode", IntegerType(), True),
    StructField("Description", StringType(), True),
    StructField("Quantity", IntegerType(), True),
    StructField("InvoiceDate", TimestampType(), True),
    StructField("UnitPrice", DoubleType(), True),
    StructField("CustomerID", IntegerType(), True),
    StructField("Country", StringType(), True)
])

retail_df = spark.read.option("header",True).schema(schema).csv(f'{base_url}/raw/online-retail')
```

```
display(retail_df)
```

| Quantity | InvoiceDate         | UnitPrice | CustomerID | Country        |
|----------|---------------------|-----------|------------|----------------|
| 1        | 2010-07-05 00:00:00 | 1.2       | 1234567890 | United Kingdom |

**Renamed all columns as per standard format using  
withColumnRenamed function**

The screenshot shows the Microsoft Azure Databricks workspace interface. A notebook titled "clean\_retail\_data" is open in the center. The code in the notebook renames several columns in a DataFrame:

```

updated_retail_column_name = retail_df.withColumnRenamed("InvoiceNo", "invoice_no") \
    .withColumnRenamed("StockCode", "stock_code") \
    .withColumnRenamed("Description", "description") \
    .withColumnRenamed("Quantity", "quantity") \
    .withColumnRenamed("InvoiceDate", "invoice_date") \
    .withColumnRenamed("UnitPrice", "unit_price") \
    .withColumnRenamed("CustomerID", "customer_id") \
    .withColumnRenamed("Country", "country")
display(updated_retail_column_name)

```

The resulting DataFrame, "updated\_retail\_column\_name", is displayed as a table with the following data:

|    | invoice_no | stock_code | description                         | quantity | invoice_date |
|----|------------|------------|-------------------------------------|----------|--------------|
| 1  | 536365     | null       | WHITE HANGING HEART T-LIGHT HOLDER  | 6        | 2010-12-01   |
| 2  | 536365     | 71053      | WHITE METAL LANTERN                 | 6        | 2010-12-01   |
| 3  | 536365     | null       | CREAM CUPID HEARTS COAT HANGER      | 8        | 2010-12-01   |
| 4  | 536365     | null       | KNITTED UNION FLAG HOT WATER BOTTLE | 6        | 2010-12-01   |
| 5  | 536365     | null       | RED WOOLLY HOTTIE WHITE HEART.      | 6        | 2010-12-01   |
| 6  | 536365     | 22752      | SET 7 BABUSHKA NESTING BOXES        | 2        | 2010-12-01   |
| 7  | 536365     | 21730      | GLASS STAR FROSTED T-LIGHT HOLDER   | 6        | 2010-12-01   |
| 8  | 536366     | 22633      | HAND WARMER UNION JACK              | 6        | 2010-12-01   |
| 9  | 536366     | 22632      | HAND WARMER RED POLKA DOT           | 6        | 2010-12-01   |
| 10 | 536367     | 84879      | ASSORTED COLOUR BIRD ORNAMENT       | 32       | 2010-12-01   |
| 11 | 536367     | 22745      | POPPY'S PLAYHOUSE BEDROOM           | 6        | 2010-12-01   |

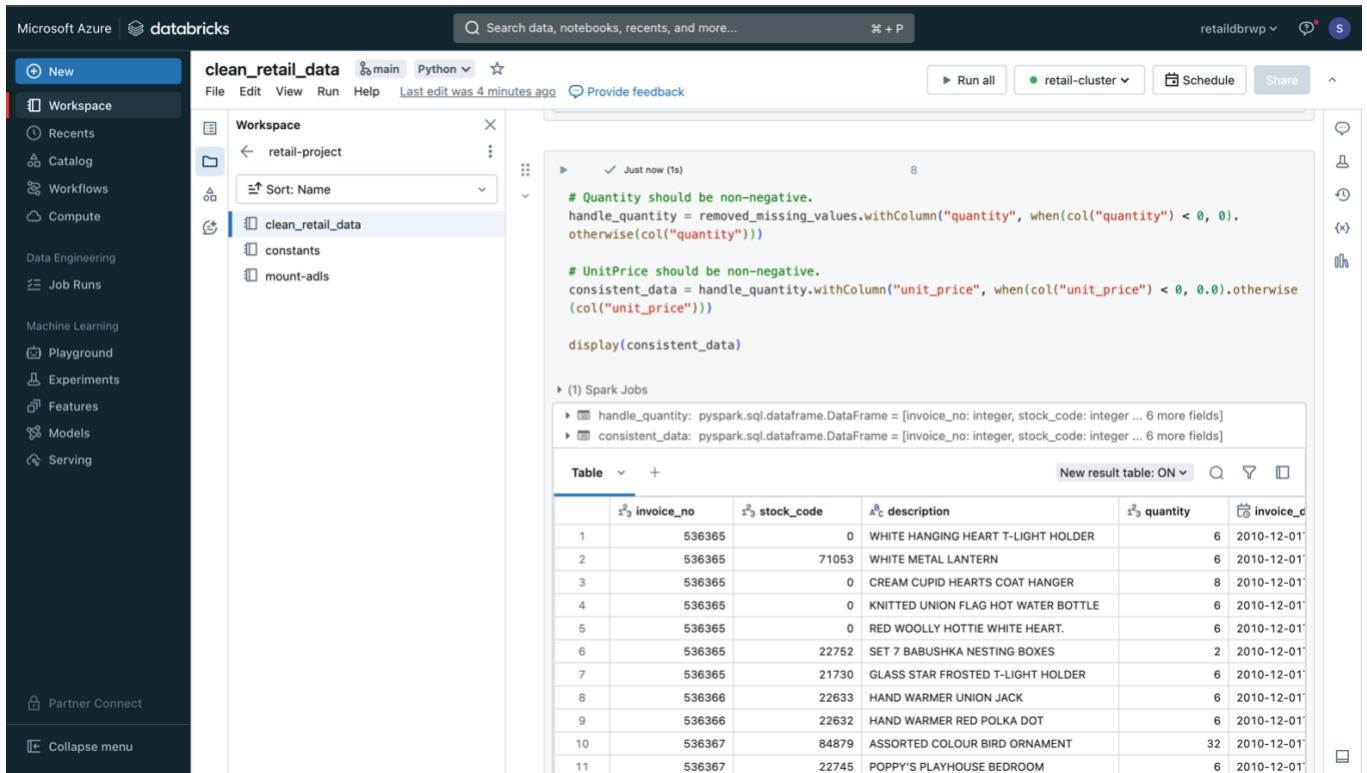
Handled all missing values in columns using Fillna function

The screenshot shows the Microsoft Azure Databricks workspace interface. On the left, the sidebar includes options like New, Workspace, Recents, Catalog, Workflows, Compute, Data Engineering, Job Runs, Machine Learning, Playground, Experiments, Features, Models, Serving, Partner Connect, and Collapse menu. The main area displays a notebook titled "clean\_retail\_data" in Python. The notebook code handles missing values by updating column names and displaying the results. The results table shows four rows of data:

|   | invoice_no | stock_code | description                         | quantity | invoice_date |
|---|------------|------------|-------------------------------------|----------|--------------|
| 1 | 536365     | 0          | WHITE HANGING HEART T-LIGHT HOLDER  | 6        | 2010-12-01   |
| 2 | 536365     | 71053      | WHITE METAL LANTERN                 | 6        | 2010-12-01   |
| 3 | 536365     | 0          | CREAM CUPID HEARTS COAT HANGER      | 8        | 2010-12-01   |
| 4 | 536365     | 0          | KNITTED UNION FLAG HOT WATER BOTTLE | 6        | 2010-12-01   |

## Correct Inconsistencies

## Removed all negative values from Quantity and Unit Price Column



The screenshot shows the Databricks workspace interface. On the left, the sidebar includes options like Workspace, Recents, Catalog, Workflows, Compute, Data Engineering, Job Runs, Machine Learning, Playground, Experiments, Features, Models, and Serving. The main area shows a notebook titled "clean\_retail\_data". The code in the notebook handles negative values for quantity and unit price:

```

# Quantity should be non-negative.
handle_quantity = removed_missing_values.withColumn("quantity", when(col("quantity") < 0, 0).otherwise(col("quantity")))

# UnitPrice should be non-negative.
consistent_data = handle_quantity.withColumn("unit_price", when(col("unit_price") < 0, 0.0).otherwise(col("unit_price")))

display(consistent_data)

```

Below the code, there's a section for "Spark Jobs" with two entries: "handle\_quantity" and "consistent\_data". A preview table is shown with the following data:

|    | invoice_no | stock_code | description                         | quantity | invoice_c  |
|----|------------|------------|-------------------------------------|----------|------------|
| 1  | 536365     | 0          | WHITE HANGING HEART T-LIGHT HOLDER  | 6        | 2010-12-01 |
| 2  | 536365     | 71053      | WHITE METAL LANTERN                 | 6        | 2010-12-01 |
| 3  | 536365     | 0          | CREAM CUPID HEARTS COAT HANGER      | 8        | 2010-12-01 |
| 4  | 536365     | 0          | KNITTED UNION FLAG HOT WATER BOTTLE | 6        | 2010-12-01 |
| 5  | 536365     | 0          | RED WOOLLY HOTTIE WHITE HEART.      | 6        | 2010-12-01 |
| 6  | 536365     | 22752      | SET 7 BABUSHKA NESTING BOXES        | 2        | 2010-12-01 |
| 7  | 536365     | 21730      | GLASS STAR FROSTED T-LIGHT HOLDER   | 6        | 2010-12-01 |
| 8  | 536366     | 22633      | HAND WARMER UNION JACK              | 6        | 2010-12-01 |
| 9  | 536366     | 22632      | HAND WARMER RED POLKA DOT           | 6        | 2010-12-01 |
| 10 | 536367     | 84879      | ASSORTED COLOUR BIRD ORNAMENT       | 32       | 2010-12-01 |
| 11 | 536367     | 22745      | POPPY'S PLAYHOUSE BEDROOM           | 6        | 2010-12-01 |

## Write Cleaned and Processed Data

The screenshot shows the Microsoft Azure Databricks workspace interface. On the left, the sidebar includes options like Workspace, Recents, Catalog, Workflows, Compute, Data Engineering, Job Runs, Machine Learning, Playground, Experiments, Features, Models, and Serving. A 'Partner Connect' section is also present.

The main area displays a DataFrame titled 'clean\_retail\_data'. The DataFrame contains 15 rows of data, with columns including ID, Price, Description, and Date. The data is sorted by Name. Below the DataFrame, it says '10,000+ rows | Truncated data due to row limit | 1.19 seconds runtime | Refreshed 4 minutes ago'.

On the right, there's a code editor cell with the following Python code:

```
new_invoice_dayofweek_column.write.mode('overwrite').parquet(f'{raw_url}/processed/retail')
```

The code editor has a status bar at the bottom indicating '[Shift+Enter] to run and move to next cell' and '[Esc H] to see all keyboard shortcuts'.

## Cleaned Data in Parquet Format in ADLS Processed Container

**Overview**

Authentication method: Access key (Switch to Microsoft Entra user account)  
Location: retail-project / processed / retail

| Name  | Modified             | Access tier    | Archive status | Blob type  | Size     | Lease state |
|---|----------------------|----------------|----------------|------------|----------|-------------|
| ...   |                      |                |                |            |          | ***         |
| _committed_2786999892612635560  | 02/06/2024, 20:41:02 | Hot (Inferred) |                | Block blob | 420 B    | Available   |
| _committed_7401785698732497983  | 03/06/2024, 03:03:45 | Hot (Inferred) |                | Block blob | 815 B    | Available   |
| _committed_7113469511618436824  | 02/06/2024, 21:05:58 | Hot (Inferred) |                | Block blob | 830 B    | Available   |
| _committed_7909668691247644908  | 03/06/2024, 03:08:53 | Hot (Inferred) |                | Block blob | 811 B    | Available   |
| _committed_vacuum6238237767287401440  | 03/06/2024, 03:03:46 | Hot (Inferred) |                | Block blob | 96 B     | Available   |
| _started_7401785698732497983  | 03/06/2024, 03:03:44 | Hot (Inferred) |                | Block blob | 0 B      | Available   |
| _started_7909668691247644908  | 03/06/2024, 03:08:51 | Hot (Inferred) |                | Block blob | 0 B      | Available   |
| part-00000-tid-7909668691247644908-f941d002-8efd-4c6e-a99f-81cb8ba00d1e-48-1.c... | 03/06/2024, 03:08:53 | Hot (Inferred) |                | Block blob | 1.8 MiB  | Available   |
| part-00001-tid-7909668691247644908-f941d002-8efd-4c6e-a99f-81cb8ba00d1e-49-1.c... | 03/06/2024, 03:08:53 | Hot (Inferred) |                | Block blob | 1.84 MiB | Available   |
| part-00002-tid-7909668691247644908-f941d002-8efd-4c6e-a99f-81cb8ba00d1e-50-1.c... | 03/06/2024, 03:08:53 | Hot (Inferred) |                | Block blob | 1.81 MiB | Available   |
| part-00003-tid-7909668691247644908-f941d002-8efd-4c6e-a99f-81cb8ba00d1e-51-1.c... | 03/06/2024, 03:08:52 | Hot (Inferred) |                | Block blob | 1.2 MiB  | Available   |

## Loaded Processed Data from ADLS

The screenshot shows the Microsoft Azure Databricks workspace interface. On the left, the sidebar includes options like Workspace, Recents, Catalog, Workflows, Compute, Data Engineering, Job Runs, Machine Learning, Playground, Experiments, Features, Models, and Serving. The main area displays a notebook titled 'transform\_retail\_data'. The notebook contains the following code:

```
from pyspark.sql.functions import *
from pyspark.sql.types import *

df = spark.read.option("header",True).parquet(f'{base_url}/processed/retail')
display(df)
```

Below the code, a table is displayed with the following data:

|   | invoice_no | stock_code | description                      | quantity | invoice_dat  |
|---|------------|------------|----------------------------------|----------|--------------|
| 1 | 549235     | 22554      | PLASTERS IN TIN WOODLAND ANIMALS | 3        | 2011-04-0771 |
| 2 | 549235     | 22557      | PLASTERS IN TIN VINTAGE PAISLEY  | 3        | 2011-04-0771 |
| 3 | 549235     | 22970      | LONDON BUS COFFEE MUG            | 6        | 2011-04-0771 |
| 4 | 549235     | 22478      | BIRDHOUSE GARDEN MARKER          | 10       | 2011-04-0771 |

## Aggregate total sales by country

The screenshot shows the Microsoft Azure Databricks workspace interface. On the left, the sidebar includes sections for Workspace, Recents, Catalog, Workflows, Compute, Data Engineering, Job Runs, Machine Learning, Playground, Experiments, Features, Models, and Serving. A 'Partner Connect' section is also present. The main area displays a notebook titled 'transform\_retail\_data' in Python. The code cell contains the following:

```
#Aggregate total sales by country
df_sales_by_country = df.groupby("country").agg(sum("total_price").alias("total_sales"))
display(df_sales_by_country)
```

The output of the code is a table titled 'Table' showing the results:

| country             | total_sales |
|---------------------|-------------|
| Sweden              | 38392       |
| Singapore           | 21297       |
| Germany             | 229580      |
| France              | 210400      |
| Greece              | 4774        |
| European Communi... | 1304        |
| Belgium             | 41360       |
| Finland             | 22611       |
| Malta               | 2735        |
| Unspecified         | 4780        |
| Italy               | 17551       |
| EIRE                | 284026      |
| Norway              | 36206       |
| Spain               | 61782       |
| Denmark             | 18962       |

The notebook status bar indicates 'Last edit was 3 minutes ago', 'Run all' button, 'retail-cluster' dropdown, 'Schedule' and 'Share' buttons, and a note about truncating data due to row limit.

## Filter Transactions with high quantities

The screenshot shows the Microsoft Azure Databricks workspace interface. On the left, there's a sidebar with various navigation options like Workspace, Recents, Catalog, Workflows, Compute, Data Engineering, Job Runs, Machine Learning, Playground, Experiments, Features, Models, and Serving. A 'Partner Connect' section is also present. At the bottom of the sidebar is a 'Collapse menu' button.

The main area displays a Python notebook titled 'transform\_retail\_data'. The notebook contains the following code:

```
#Filter transactions with high quantities
df_high_quantity = df.filter(col("quantity") > 100)
display(df_high_quantity)
```

The notebook has run successfully, indicated by a green checkmark icon. The output shows a table with 15 rows of data, each containing columns: invoice\_no, stock\_code, description, quantity, and invoice\_date. The data includes various retail items like 'PLACE SETTING WHITE HEART' and 'PACK OF 6 PANNETONE GIFT BOXES'.

| invoice_no | stock_code | description                        | quantity | invoice_date |
|------------|------------|------------------------------------|----------|--------------|
| 1          | 549246     | PLACE SETTING WHITE HEART          | 300      | 2011-04-07T1 |
| 2          | 549246     | PACK OF 6 PANNETONE GIFT BOXES     | 144      | 2011-04-07T1 |
| 3          | 549255     | HANGING JAM JAR T-LIGHT HOLDER     | 192      | 2011-04-07T1 |
| 4          | 549412     | "ASSORTED FLOWER COLOUR ""LEIS"""  | 1200     | 2011-04-08T1 |
| 5          | 549431     | MINI PAINT SET VINTAGE             | 180      | 2011-04-08T1 |
| 6          | 549447     | HANGING JAM JAR T-LIGHT HOLDER     | 192      | 2011-04-08T1 |
| 7          | 549520     | ASSORTED COLOUR BIRD ORNAMENT      | 160      | 2011-04-08T1 |
| 8          | 549541     | HANGING JAM JAR T-LIGHT HOLDER     | 192      | 2011-04-10T1 |
| 9          | 549590     | PACK OF 72 SKULL CAKE CASES        | 120      | 2011-04-11T1 |
| 10         | 549590     | 60 TEATIME FAIRY CAKE CASES        | 120      | 2011-04-11T1 |
| 11         | 549590     | PACK OF 60 PINK PAISLEY CAKE CASES | 120      | 2011-04-11T1 |
| 12         | 549591     | PAPER CHAIN KIT EMPIRE             | 330      | 2011-04-11T1 |
| 13         | 549593     | 20 DOLLY PEGS RETROSPOT            | 130      | 2011-04-11T1 |
| 14         | 549595     | 200 BENDY SKULL STRAWS             | 192      | 2011-04-11T1 |
| 15         | 549595     | PACK OF 72 SKULL CAKE CASES        | 120      | 2011-04-11T1 |

At the bottom of the notebook, it says '4,950 rows | 0.71 seconds runtime' and 'Refreshed now'.

## Calculate Average Unit Price by Stock Code

The screenshot shows the Microsoft Azure Databricks workspace interface. On the left, the sidebar includes sections for Workspace, Recents, Catalog, Workflows, Compute, Data Engineering, Job Runs, Machine Learning, Playground, Experiments, Features, Models, and Serving. A 'Partner Connect' section is also present. The main area displays a notebook titled 'transform\_retail\_data' in Python. The notebook contains the following code:

```
# Calculate average unit price by stock code
df_avg_price_by_stock = df.groupby("stock_code").agg(round(avg("unit_price"), 2).alias("avg_unit_price"))

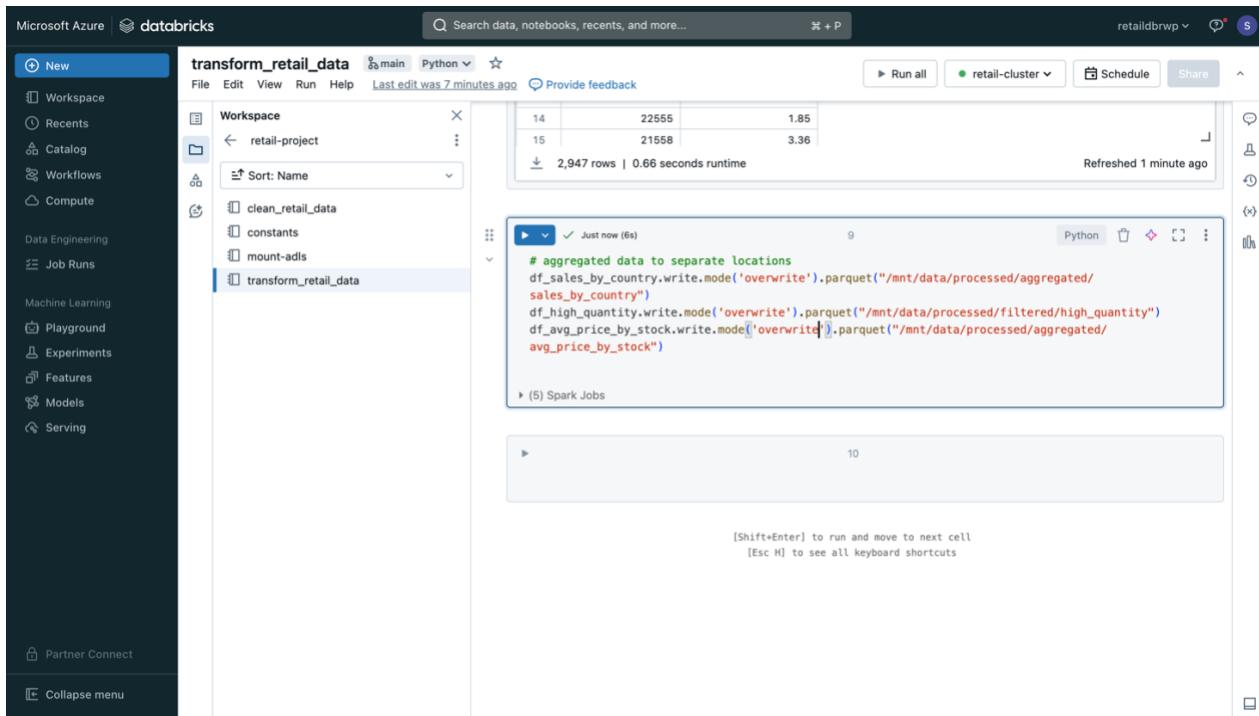
display(df_avg_price_by_stock)
```

The notebook has run once, with a timestamp of "Just now (1s)". The resulting table is displayed below:

| 1 stock_code | 1.2 avg_unit_price |
|--------------|--------------------|
| 1            | 0.89               |
| 2            | 0.68               |
| 3            | 2.39               |
| 4            | 3.88               |
| 5            | 1.1                |
| 6            | 1.55               |
| 7            | 0.59               |
| 8            | 5.3                |
| 9            | 0.81               |
| 10           | 1.05               |
| 11           | 2.6                |
| 12           | 8.61               |
| 13           | 2.72               |
| 14           | 1.85               |
| 15           | 3.36               |

Below the table, it says "2,947 rows | 0.66 seconds runtime". The notebook was "Refreshed now".

## Write Aggregated Data in different Directory

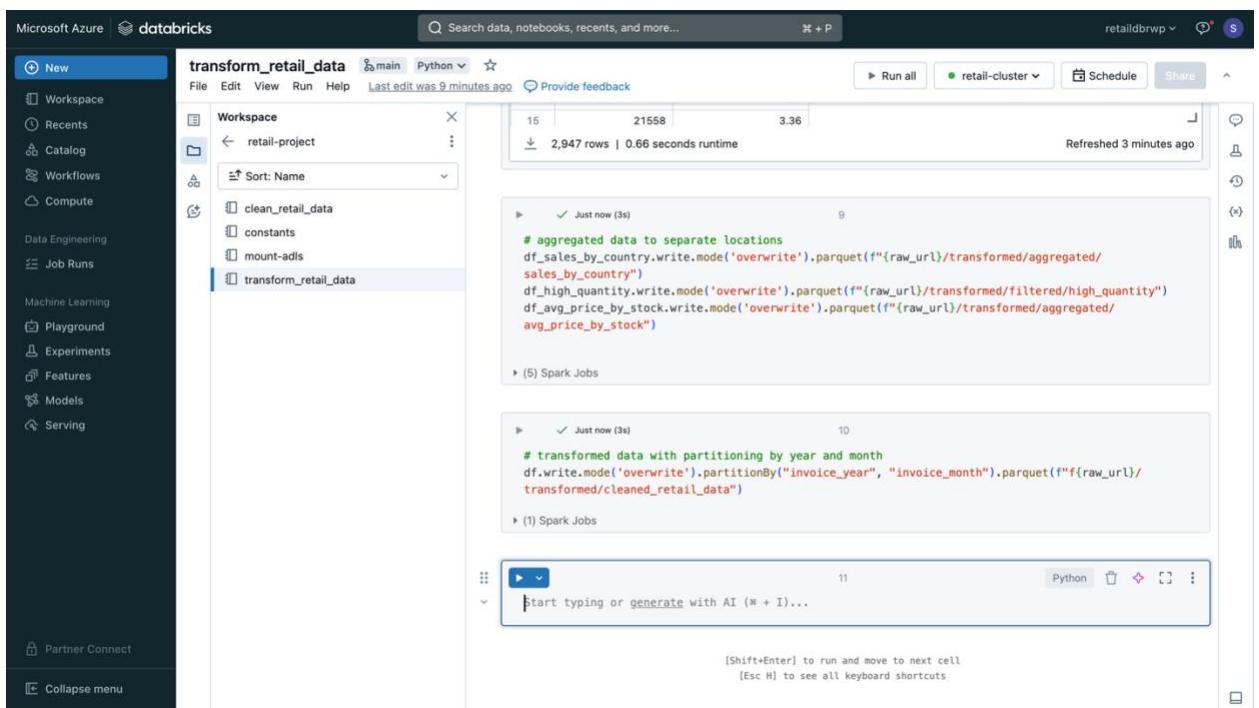


The screenshot shows a Databricks notebook titled "transform\_retail\_data". The notebook interface includes a sidebar with various workspace options like Workspace, Recents, Catalog, Workflows, Compute, and Machine Learning. The main area displays a table with three columns: 14, 22555, and 1.85. Below the table, a code cell shows Python code for writing aggregated data to separate locations:

```
# aggregated data to separate locations
df_sales_by_country.write.mode('overwrite').parquet("/mnt/data/processed/aggregated/sales_by_country")
df_high_quantity.write.mode('overwrite').parquet("/mnt/data/processed/filtered/high_quantity")
df_avg_price_by_stock.write.mode('overwrite').parquet("/mnt/data/processed/aggregated/avg_price_by_stock")
```

The notebook also shows a section for "Spark Jobs" and a status bar indicating "Refreshed 1 minute ago".

## Write Aggregated Data with Partition by Year and Month Directory



This screenshot shows the same Databricks notebook environment. The code cell now includes additional logic for partitioning the data by year and month:

```
# aggregated data to separate locations
df_sales_by_country.write.mode('overwrite').parquet(f"{raw_url}/transformed/aggregated/sales_by_country")
df_high_quantity.write.mode('overwrite').parquet(f"{raw_url}/transformed/filtered/high_quantity")
df_avg_price_by_stock.write.mode('overwrite').parquet(f"{raw_url}/transformed/aggregated/avg_price_by_stock")

# transformed data with partitioning by year and month
df.write.mode('overwrite').partitionBy("invoice_year", "invoice_month").parquet(f"{raw_url}/transformed/cleaned_retail_data")
```

The notebook shows a "Spark Jobs" section and a status bar indicating "Refreshed 3 minutes ago".

## Aggregated and Filtered Data in ADLS

The screenshot shows the Microsoft Azure Storage Explorer interface. At the top, there's a navigation bar with 'Microsoft Azure' and an 'Upgrade' button. A search bar says 'Search resources, services, and docs (G+ /)'. On the right, it shows the email 'Shrji74@outlook.com' and 'DEFAULT DIRECTORY'. Below the navigation, the path 'Home > retailads | Containers >' leads to the 'retail-project' container. The container has a 'Container' icon and the name 'retail-project'. The left sidebar has a 'Overview' tab selected, along with 'Diagnose and solve problems', 'Access Control (IAM)', and 'Settings'. The main area shows a table of blobs:

| Name       | Modified | Access tier | Archive status | Blob type | Size | Lease state |
|------------|----------|-------------|----------------|-----------|------|-------------|
| aggregated |          |             |                |           | -    | ...         |
| filtered   |          |             |                |           | -    | ...         |

There are also buttons for 'Upload', 'Add Directory', 'Refresh', 'Rename', 'Delete', 'Change tier', 'Acquire lease', 'Break lease', and 'Give feedback'. A search bar at the top says 'Search blobs by prefix (case-sensitive)' and a toggle switch says 'Show deleted objects'.

## Creating Linked Service of Databricks in ADF using Access Token

The screenshot shows the Microsoft Azure Data Factory interface for creating a new linked service. On the left, the 'Factory Resources' sidebar lists Pipelines, Datasets, Data flows, Power Query, and Templates. In the center, a pipeline named 'ds\_retail\_adls\_output' is selected. The 'Activities' pane shows various options like Move and transform, Synapse, Azure Data Explorer, Azure Function, Batch Service, and Databricks. Under Databricks, there are Notebook, Jar, and Python options. The main panel is titled 'New linked service' and is configured for 'Azure Databricks'. The 'Account selection method' is set to 'From Azure subscription' (radio button selected). The 'Azure subscription' dropdown shows 'Shriji-Wp (9b093c33-a116-4e12-aa43-073a1c622e60)'. The 'Databricks workspace' dropdown shows 'retaildbwpp'. Under 'Select cluster', the 'Existing interactive cluster' radio button is selected. The 'Databrick Workspace URL' field contains 'https://adb-3107986375698214.14.azuredatabricks.net'. The 'Authentication type' is set to 'Access Token', and the 'Access token' field contains a redacted value. The 'Choose from existing clusters' dropdown shows 'retail-cluster'. At the bottom right, a green checkmark icon indicates 'Connection successful' and a blue link says 'Test connection'. There are 'Create' and 'Cancel' buttons at the bottom.

## Creating Pipeline of Databricks notebook in ADF

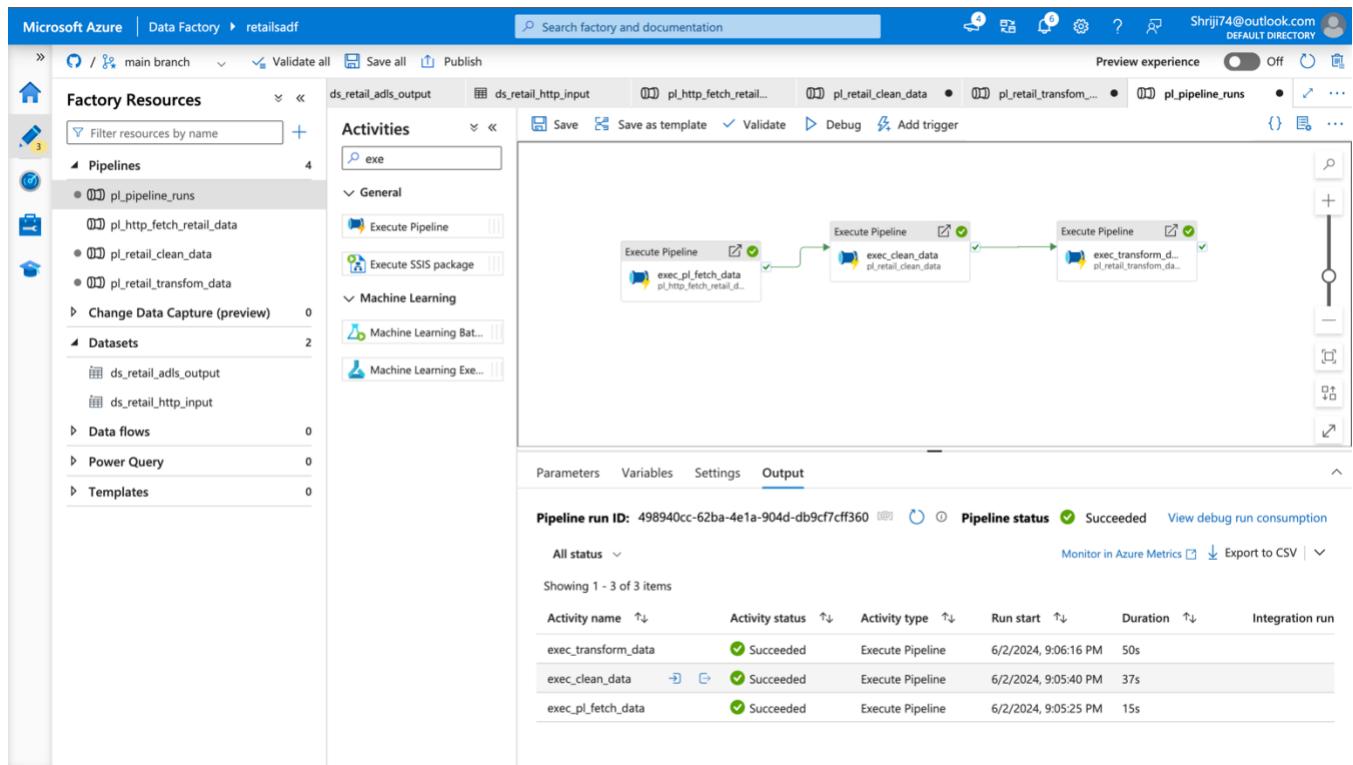
### Clean Data Notebook

The screenshot shows the Microsoft Azure Data Factory pipeline editor. On the left, the 'Factory Resources' sidebar lists several pipelines, datasets, and other resources. In the main workspace, a pipeline named 'pl\_retail\_transform\_data' is selected. Within this pipeline, there is a single activity named 'clean\_notebook'. The 'Settings' tab is active for this activity, showing the 'Notebook path' as '/Repos/shriji74@outlook.com/project-2...'. The pipeline editor interface includes various buttons for saving, validating, debugging, and publishing.

### Transformed Data Notebook

This screenshot shows the same Microsoft Azure Data Factory pipeline editor interface as the previous one. The pipeline 'pl\_retail\_transform\_data' is still selected. However, the activity 'clean\_notebook' has been replaced by a new activity named 'transformed\_notebook'. The 'Settings' tab for this new activity is also active, showing the same 'Notebook path' as the previous activity. The pipeline editor interface remains consistent with the first screenshot.

## Creating Pipeline to run three pipelines of Fetch Data, Clean Data Notebook and Transformed Data Notebook



# Azure Synapse Analytics

## Creating Synapse Workspace

**Overview**

**Resource group:** health-synapse

**Status:** Succeeded

**Location:** East US

**Subscription:** Shreeji-Wo

**Managed virtual network:** No

**Managed identity object ID:** 2351eeaa-6988-4389-8ab9-c0b49bcb7d9c

**Workspace web URL:** <https://web.azure-synapse.net/workspace=%2fsubscriptions%2f09093c33->

**Tags:** (empty) Add tags

**Getting started**

- Open Synapse Studio**: Start building your fully-integrated analytics solution and unlock new insights. [Open](#)
- Read documentation**: Learn how to be productive quickly. Explore concepts, tutorials, and samples. [Learn more](#)

**Analytics pools**

| Name                 | Type       | Size   |
|----------------------|------------|--------|
| SQL pools            |            |        |
| Built-in             | Serverless | Auto   |
| shriji               | Dedicated  | DW100c |
| Apache Spark pools   |            |        |
| No pools provisioned |            |        |
| Data Explorer pools  |            |        |
| No pools provisioned |            |        |

Page 1 of 1

## Git Integration

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. The left sidebar has a 'Manage' section selected. The main area is titled 'Git repository' and displays configuration details for a GitHub repository named 'project\_1'. Key settings include:

- Repository type:** GitHub
- GitHub account:** meet74
- Repository name:** project\_1
- Collaboration branch:** main
- Publish branch:** adf\_publish
- Root folder:** /
- Last published commit:** a1b32c0c4a808b07befb3dd21159d47407dda1f4
- Custom comment:** Enabled

## Creating Linked Service for Azure Synapse Analytics in ADF

The screenshot shows the Microsoft Azure Data Factory workspace. The left sidebar has a 'Linked services' section selected. The main area shows a list of existing linked services and a form for creating a new one. The new linked service is being configured with the following details:

- Name:** ls\_synapse
- Type:** Azure Synapse Analytics
- Connect via integration runtime:** AutoResolveIntegrationRuntime
- Version:** Recommended
- Account selection method:** Enter manually
- Fully qualified domain name:** health-synapse.sql.azuresynapse.net
- Database name:** shriji
- Authentication type:** SQL authentication
- User name:** shriji
- Password:** (Azure Key Vault)

## Creating Table in Dedicated SQL Pool

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. The left sidebar has 'Data' selected under 'Workspace'. In the center, there are three tabs: 'SQL script 1', 'SQL script 2', and 'SQL script 3'. 'SQL script 3' is active, displaying the following SQL code:

```
1 CREATE TABLE cleaned_retail_data (
2     invoice_no NVARCHAR(100),
3     stock_code NVARCHAR(100),
4     description NVARCHAR(255),
5     quantity INT,
6     invoice_date DATETIME,
7     unit_price FLOAT,
8     customer_id INT,
9     country NVARCHAR(100),
10    total_price FLOAT,
11    transaction_id NVARCHAR(100),
12    invoice_year INT,
13    invoice_month INT,
14    invoice_day_of_week NVARCHAR(100)
15
16 WITH(
17     DISTRIBUTION = ROUND_ROBIN
18 )
```

The right side shows the 'Properties' panel for 'SQL script 3', with 'Name' set to 'SQL script 3'. Below the code, it says 'No results to show' and 'Your query yielded no displayable results'. At the bottom, it says '00:00:02 Query executed successfully.'

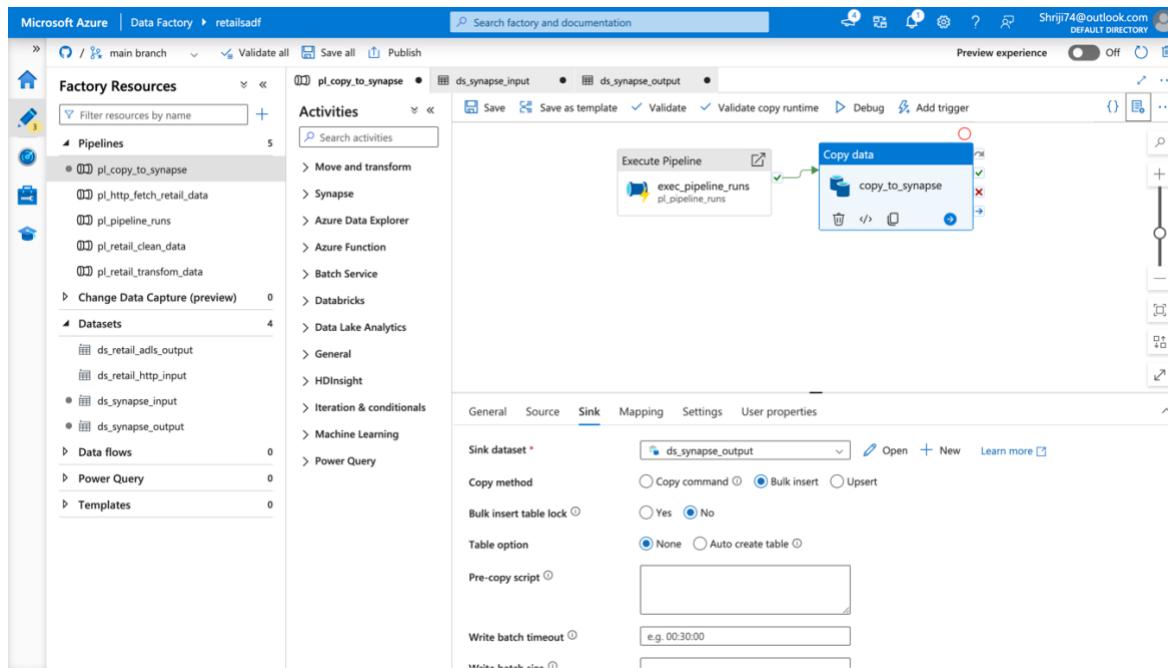
## Creating Dataset for Synapse in ADF

The screenshot shows the Microsoft Azure Data Factory interface. On the left, the 'Factory Resources' sidebar lists Pipelines, Datasets, Data flows, Power Query, and Templates. Under Pipelines, 'pl\_copy\_to\_synapse' is selected. Under Datasets, 'ds\_synapse\_input' is selected. The main workspace displays the 'Set properties' dialog for 'ds\_synapse\_input'. The 'Source' tab is active, showing the configuration for the dataset:

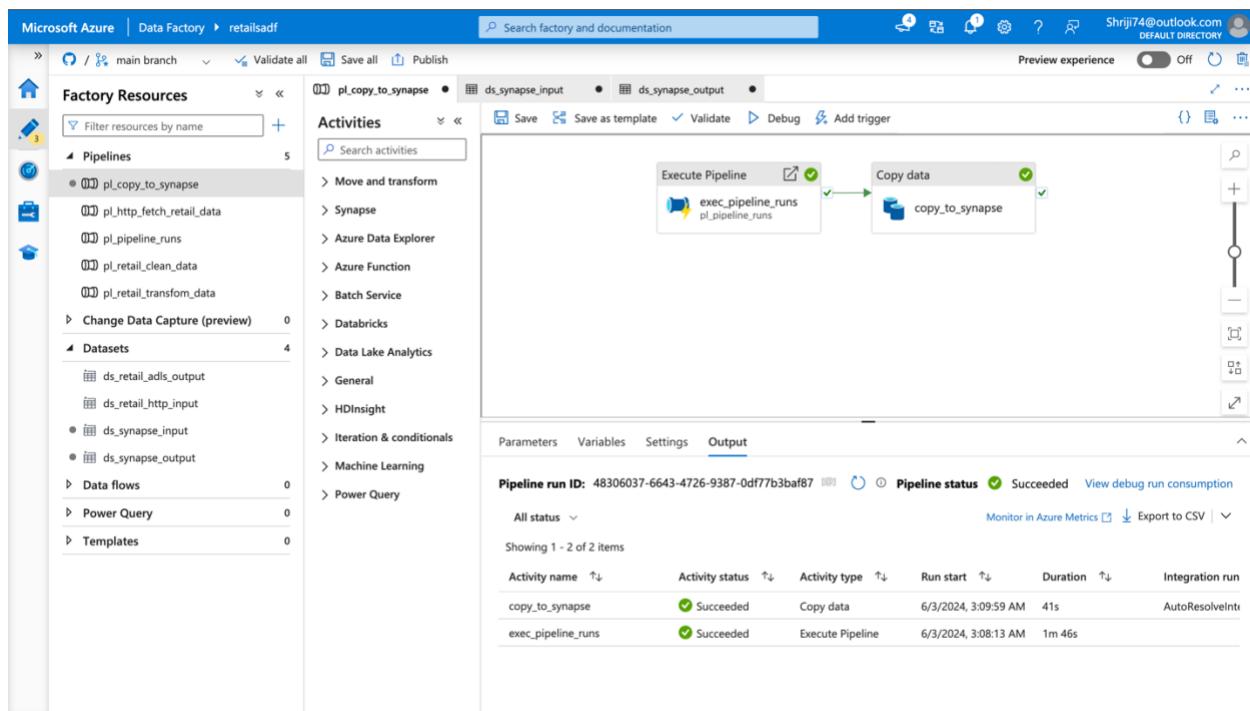
- Name: ds\_synapse\_output
- Linked service: ls\_synapse
- Table name: dbo.cleaned\_retail\_data
- Import schema: From connection/store (selected)
- Source dataset: ds (selected)
- File path type: File (selected)
- Filter by last modified: Start time (UTC) (empty)
- Recursively: checked
- Enable partitions discovery: unchecked

At the bottom right of the dialog are 'OK', 'Back', and 'Cancel' buttons.

## Creating a Pipeline that executes after Data Transformation and Loading Data in Dedicated SQL Pool Table *Cleaned\_data*



**Pipeline Succeeded**



## Loaded Data in SQL Dedicated Pool

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. The left sidebar displays 'Data' resources, including a workspace named 'shriji'. The main area is a 'SQL script 3' editor. The script content is as follows:

```
5     description NVARCHAR(255),
6     quantity INT,
7     invoice_date DATETIME,
8     unit_price FLOAT,
9     customer_id INT,
10    country NVARCHAR(100),
11    total_price FLOAT,
12    transaction_id NVARCHAR(100),
13    invoice_year INT,
14    invoice_month INT,
15    invoice_dayof_week NVARCHAR(100)
16  )  
WITH(  
17    DISTRIBUTION = ROUND_ROBIN  
18 );  
19  
20  
21  
22 SELECT * FROM cleaned_retail_data
```

The 'Results' tab is selected, showing a table with the following data:

| invoice_no | stock_code | description     | quantity | invoice_date      | unit_price | custo |
|------------|------------|-----------------|----------|-------------------|------------|-------|
| 574942     | 21915      | RED HARMONI...  | 240      | 2011-11-07T17:... | 1.06       | 1533€ |
| 574950     | 22195      | LARGE HEART ... | 1        | 2011-11-08T09:... | 3.29       | 0     |
| 536401     | 0          | SCANDINAVIA...  | 2        | 2010-12-01T11:... | 1.25       | 1586€ |
| 536404     | 20728      | LUNCH BAG CA... | 10       | 2010-12-01T11:... | 1.65       | 1621€ |

A message at the bottom indicates: '00:00:03 Query executed successfully.'

## Performing Different Analysis in Dedicated SQL Pool

### Total Sales by Country

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. The left sidebar displays 'Data' resources, including 'Workspace' and 'SQL database'. The main area shows two SQL scripts: 'SQL script 3' and 'SQL script 4'. 'SQL script 4' contains the following T-SQL code:

```
1 -- Total sales by country
2 SELECT country, SUM(total_price) AS total_sales
3 FROM cleaned_retail_data
4 GROUP BY country
5 ORDER BY total_sales DESC;
```

The results of the query are displayed in a table:

| country        | total_sales |
|----------------|-------------|
| United Kingdom | 9049148     |
| Netherlands    | 285458      |
| EIRE           | 284026      |
| Germany        | 229580      |

A message at the bottom indicates: "00:00:02 Query executed successfully."

The right side of the screen shows the 'Properties' panel for 'SQL script 4', which includes fields for Name (set to 'SQL script 4'), Description, Type (.sql script), Size (0 bytes), and Results settings per query (set to 'First 5000 rows (default)').

## Average Unit price by Stock Code

```

13  -- Average unit price by stock code
14  CREATE TABLE Average_Price_Stock
15  WITH(
16      |   DISTRIBUTION = ROUND_ROBIN
17  )
18  AS
19  SELECT stock_code, ROUND(AVG(unit_price), 2) AS avg_unit_price
20  FROM cleaned_retail_data
21  GROUP BY stock_code
22
23
24
--
```

## Monthly Sales Trends

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. On the left, there's a sidebar with icons for Home, Databases, Workspaces, and Pipelines. The main area has tabs for Data, Workspace, and Linked. Under Data, there's a 'SQL database' section with a dropdown for 'shriji (SQL)'.

Two SQL scripts are visible in the center:

- SQL script 3:**

```

1  SELECT country, sum(total_price) AS total_sales
2  FROM cleaned_retail_data
3  GROUP BY country
4  ORDER BY total_sales DESC;
5
6
7
8  -- Avgag Name - stock_code
9  SELECT st.Type - nvarchar ) AS avg_unit_price
10 FROM clea Scope - dbo.cleaned_retail_data
11 GROUP BY stock_code
12 ORDER BY avg_unit_price DESC;
13
14
15  [- Monthly sales trends
16  SELECT invoice_year, invoice_month, SUM(total_price) AS total_sales
17  FROM cleaned_retail_data
18  GROUP BY invoice_year, invoice_month
19  ORDER BY invoice_year, invoice_month;
20
```
- SQL script 4:**

```

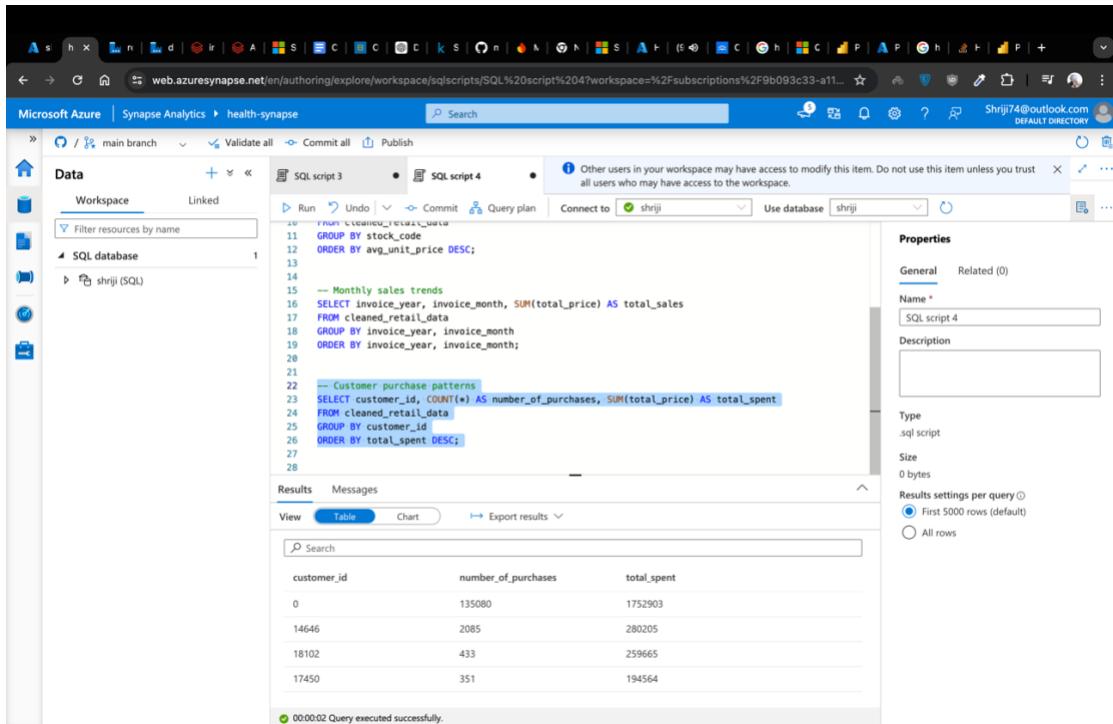
1  Other users in your workspace may have access to modify this item. Do not use this item unless you trust all users who may have access to the workspace.
2
3  Connect to shriji Use database shriji
```

The results for the first query are displayed in a table:

| invoice_year | invoice_month | total_sales |
|--------------|---------------|-------------|
| 2010         | 12            | 825933      |
| 2011         | 1             | 693138      |
| 2011         | 2             | 525224      |
| 2011         | 3             | 719758      |

At the bottom, a message says "000002 Query executed successfully."

## Customer Purchase Patterns



The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. The left sidebar shows 'Data' selected under 'Workspace'. The main area displays a SQL script named 'SQL script 4' with the following code:

```

11 -- Monthly sales trends
12 SELECT invoice_year, invoice_month, SUM(total_price) AS total_sales
13 FROM cleaned_retail_data
14 GROUP BY invoice_year, invoice_month
15 ORDER BY invoice_year, invoice_month;
16
17 -- Customer purchase patterns
18 SELECT customer_id, COUNT(*) AS number_of_purchases, SUM(total_price) AS total_spent
19 FROM cleaned_retail_data
20 GROUP BY customer_id
21 ORDER BY total_spent DESC;
22
23 -- Transactions with high quantities
24 SELECT invoice_no, stock_code, quantity, total_price
25 FROM cleaned_retail_data
26 WHERE quantity > 100
27 ORDER BY quantity DESC;
28
29 -- Transactions with high quantities
30 SELECT invoice_no, stock_code, quantity, total_price
31 FROM cleaned_retail_data
32 WHERE quantity > 100
33 ORDER BY quantity DESC;
34
35

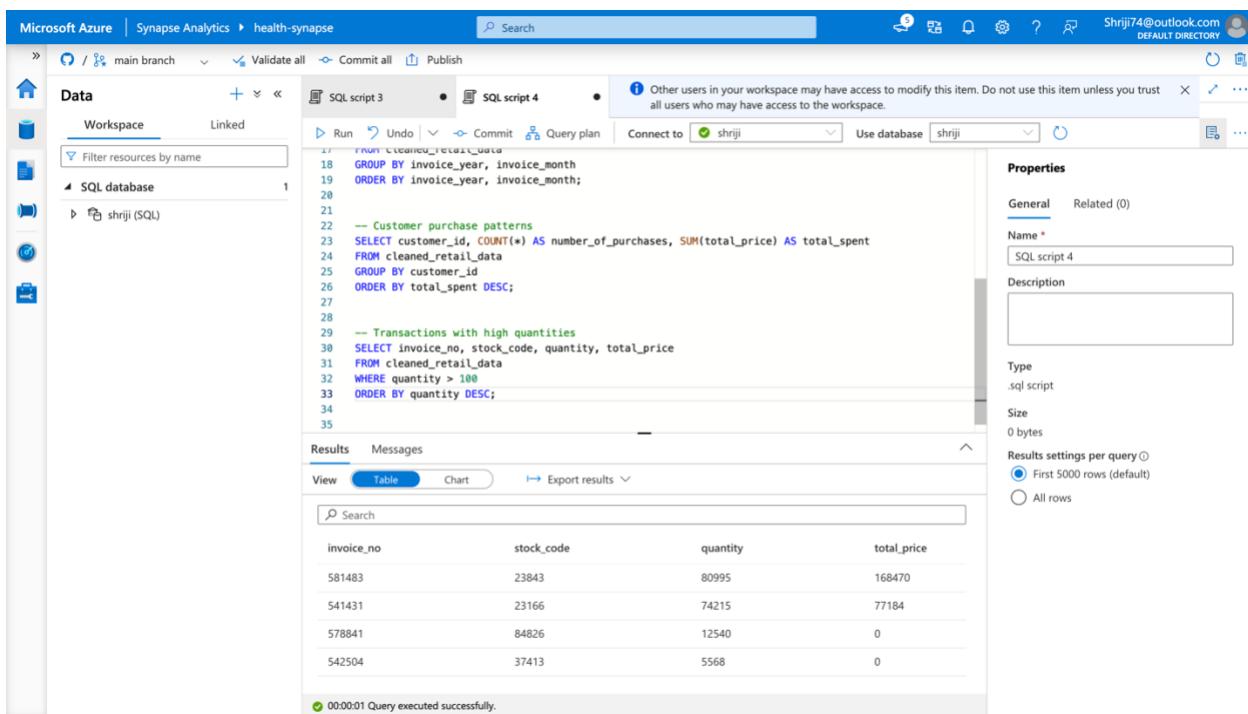
```

The results pane shows a table with three columns: customer\_id, number\_of\_purchases, and total\_spent. The data is as follows:

| customer_id | number_of_purchases | total_spent |
|-------------|---------------------|-------------|
| 0           | 135080              | 1752903     |
| 14646       | 2085                | 280205      |
| 18102       | 433                 | 259665      |
| 17450       | 351                 | 194564      |

A message at the bottom indicates '00:00:02 Query executed successfully.'

## Transactions High Quantities



The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. The left sidebar shows 'Data' selected under 'Workspace'. The main area displays a SQL script named 'SQL script 4' with the following code:

```

11 -- Monthly sales trends
12 SELECT invoice_year, invoice_month, SUM(total_price) AS total_sales
13 FROM cleaned_retail_data
14 GROUP BY invoice_year, invoice_month
15 ORDER BY invoice_year, invoice_month;
16
17 -- Customer purchase patterns
18 SELECT customer_id, COUNT(*) AS number_of_purchases, SUM(total_price) AS total_spent
19 FROM cleaned_retail_data
20 GROUP BY customer_id
21 ORDER BY total_spent DESC;
22
23 -- Transactions with high quantities
24 SELECT invoice_no, stock_code, quantity, total_price
25 FROM cleaned_retail_data
26 WHERE quantity > 100
27 ORDER BY quantity DESC;
28
29 -- Transactions with high quantities
30 SELECT invoice_no, stock_code, quantity, total_price
31 FROM cleaned_retail_data
32 WHERE quantity > 100
33 ORDER BY quantity DESC;
34
35

```

The results pane shows a table with four columns: invoice\_no, stock\_code, quantity, and total\_price. The data is as follows:

| invoice_no | stock_code | quantity | total_price |
|------------|------------|----------|-------------|
| S81483     | 23843      | 80995    | 168470      |
| S41431     | 23166      | 74215    | 77184       |
| S78841     | 84826      | 12540    | 0           |
| S42504     | 37413      | 5568     | 0           |

A message at the bottom indicates '00:00:01 Query executed successfully.'

## Creating Tables of all this Analysis for Visualization

The screenshot shows the Microsoft Azure Synapse Analytics workspace interface. The left sidebar displays 'Develop' resources, including 'SQL scripts' (SQL script 1, SQL script 2, SQL script 3, SQL script 4) and 'Power BI'. The main area shows a SQL editor with two scripts. Script 3 is a comment block, and Script 4 contains the following T-SQL code:

```
1 -- Total sales by country
2 CREATE TABLE Total_sales_country
3 WITH(
4     DISTRIBUTION = ROUND_ROBIN
5 )
6 AS
7 SELECT country, SUM(total_price) AS total_sales
8 FROM cleaned_retail_data
9 GROUP BY country
10
11
12
13 -- Average unit price by stock code
14 CREATE TABLE Average_Price_Stock
15 WITH(
16     DISTRIBUTION = ROUND_ROBIN
17 )
18 AS
19 SELECT stock_code, ROUND(AVG(unit_price), 2) AS avg_unit_price
```

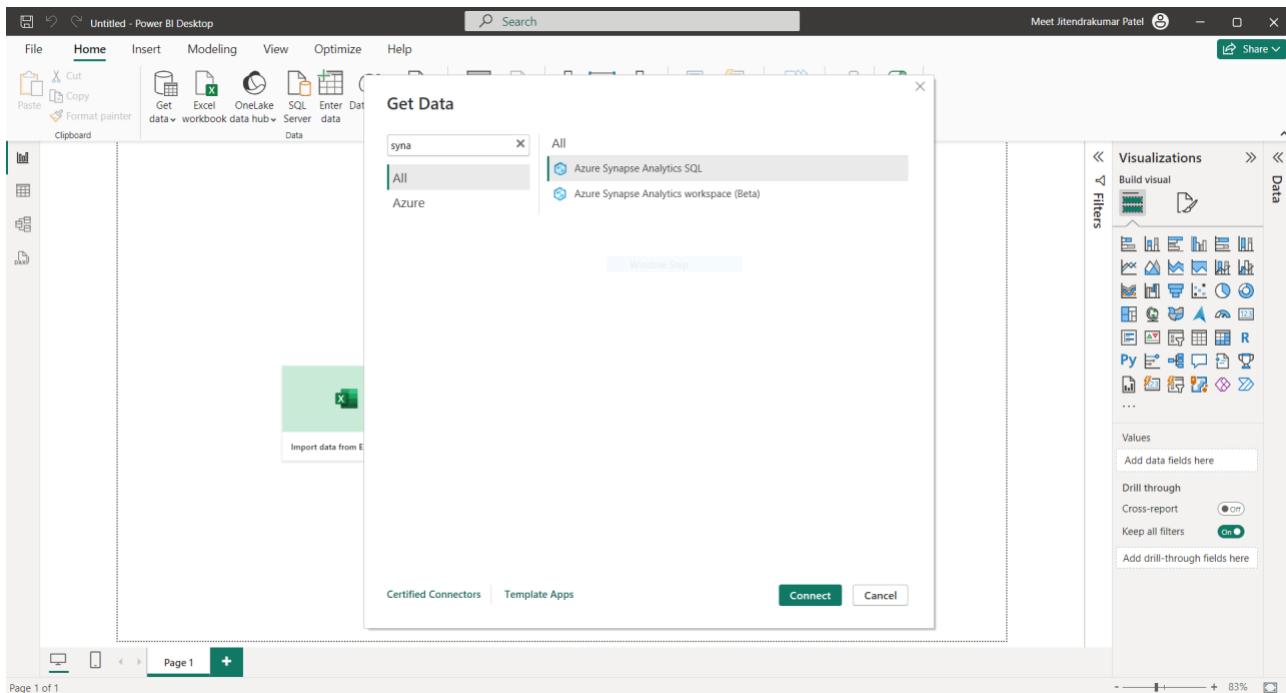
The 'Properties' panel on the right shows the following details for 'SQL script 4':

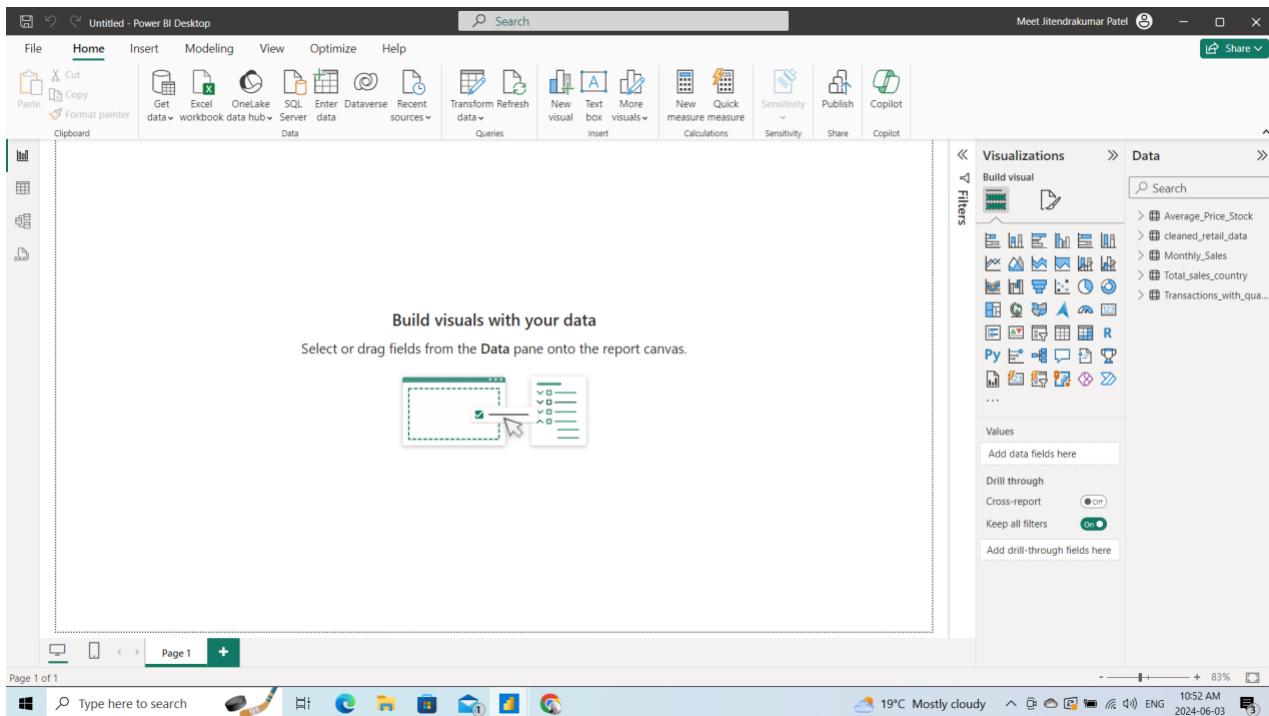
- Name: SQL script 4
- Type: .sql script
- Size: 0 bytes
- Results settings per query:
  - First 5000 rows (default)
  - All rows

The 'Results' tab indicates 'No results to show' and 'Your query yielded no displayable results'. A status bar at the bottom says '00:00:04 Query executed successfully.'

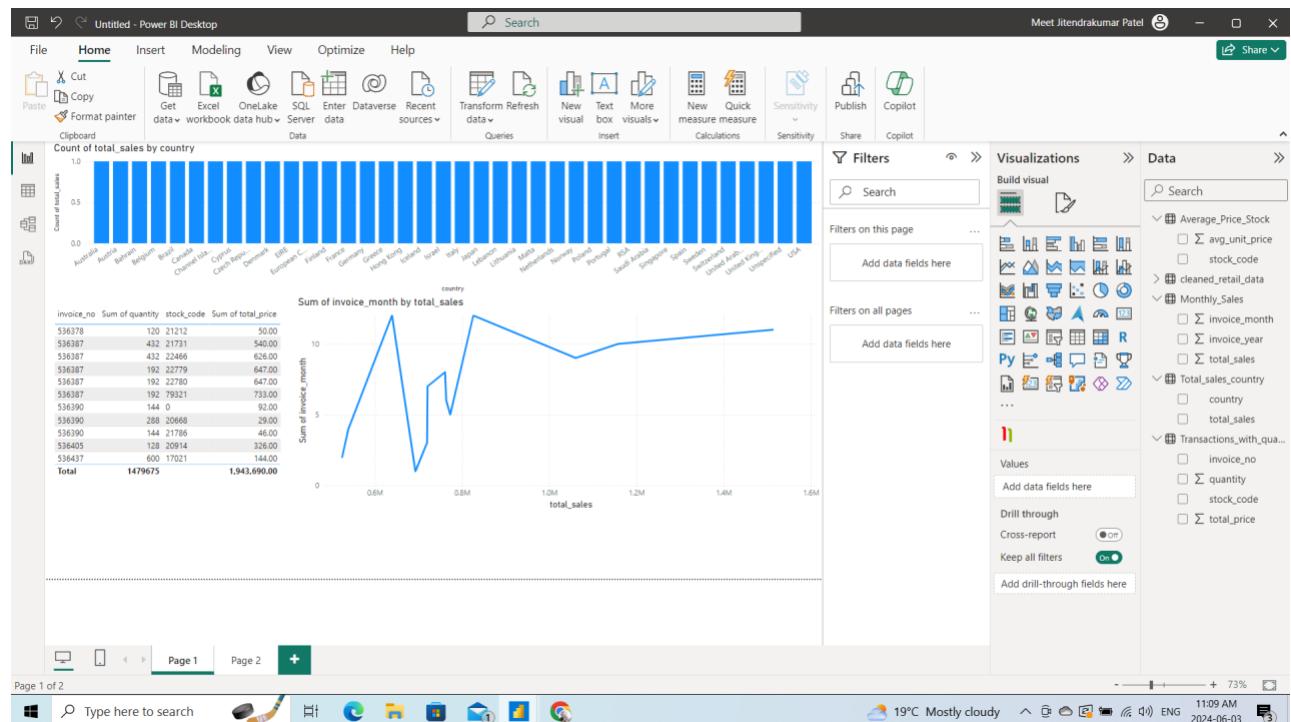
# PowerBI

## Linking PowerBI with Synapse Analytics





## Creating Different Visualization using different Charts



## Conclusion

My project demonstrated a robust approach to building a comprehensive data pipeline, which significantly improved inventory management and sales forecasting for the sales department. Insights from the survey provided information a provided value to support strategic decision making. Implementation of Azure services such as Data Factory, Data Lake Storage, Databricks, Synapse Analytics, and Power BI ensured the pipeline was efficient, effective, and efficient.

By integrating data sources, performing thorough data cleaning and transformation, and comprehensive analysis, the project successfully met its objectives. The result was actionable insights that provided company the retailer was able to make informed decisions, operate efficiently and improve overall efficiency.

## GitHub

[Github Link](#)