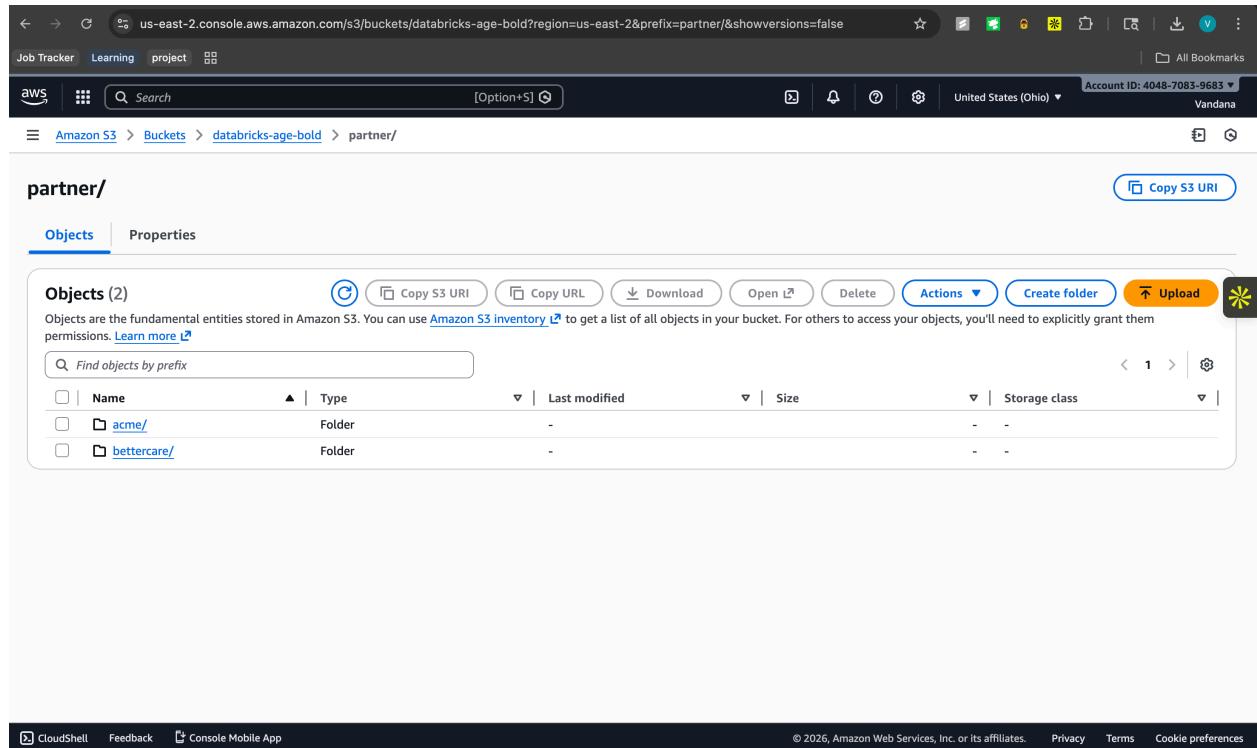


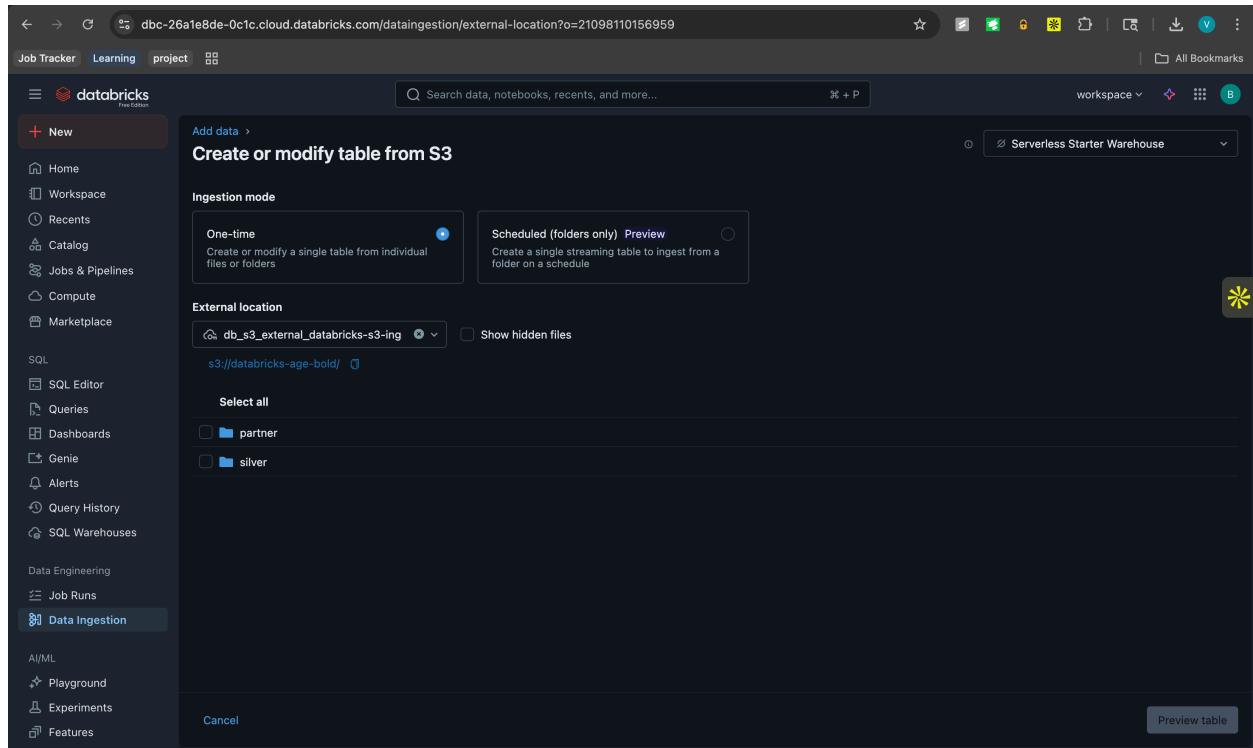
1. Uploaded the files into S3



The screenshot shows the AWS S3 console interface. The URL in the address bar is `us-east-2.console.aws.amazon.com/s3/buckets/databricks-age-bold?region=us-east-2&prefix=partner/&showversions=false`. The page displays a list of objects in the 'partner' folder of the 'databricks-age-bold' bucket. There are two items: 'acme/' and 'bettercare/'. Both are listed as 'Folder' type objects. The table has columns for Name, Type, Last modified, Size, and Storage class. The 'Actions' button is visible at the top right of the table. The bottom of the screen shows standard AWS navigation links like CloudShell, Feedback, and Console Mobile App, along with copyright and privacy information.

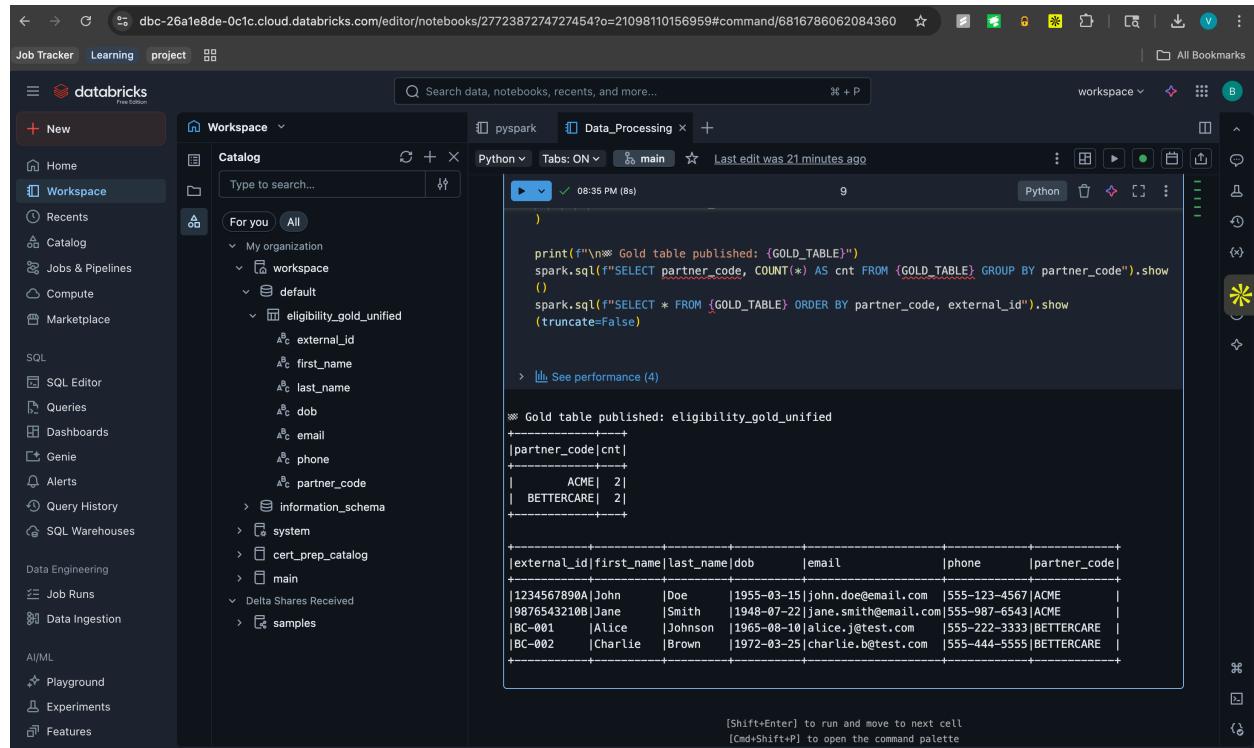
Name	Type	Last modified	Size	Storage class
acme/	Folder	-	-	-
bettercare/	Folder	-	-	-

2. Connected the Databricks to S3 using External S3 location and added files in it as raw layer



The screenshot shows the Databricks Data Ingestion interface. The left sidebar is dark-themed and includes sections for Home, Workspace, Recents, Catalog, Jobs & Pipelines, Compute, Marketplace, SQL, and Data Engineering. The 'Data Ingestion' section is highlighted with a dark background. The main content area is titled 'Create or modify table from S3' and shows the 'Ingestion mode' section with 'One-time' selected. It also shows the 'External location' section with 'db_s3_external_databricks-s3-ing' selected and 'partner' and 'silver' folders listed under 'Select all'. A 'Preview table' button is visible at the bottom right.

3. Queried the data from ingested files into dataframes and created a final curated table for downstream applications



The screenshot shows the Databricks workspace interface. On the left, the sidebar includes sections for Home, Workspace, Recents, Catalog, Jobs & Pipelines, Compute, Marketplace, SQL, SQL Editor, Queries, Dashboards, Genie, Alerts, Query History, SQL Warehouses, Data Engineering, Job Runs, Data Ingestion, AI/ML, Playground, Experiments, and Features. The 'Workspace' tab is selected.

The main area displays a 'Data_Processing' notebook with a single Python cell. The cell content is as follows:

```
print(f"\n** Gold table published: {GOLD_TABLE}")
spark.sql("SELECT partner_code, COUNT(*) AS cnt FROM {GOLD_TABLE} GROUP BY partner_code").show()
spark.sql(f"SELECT * FROM {GOLD_TABLE} ORDER BY partner_code, external_id").show(truncate=False)
```

The output of the first command shows the count of rows for each partner code:

partner_code	cnt
ACME	2
BETTERCARE	2

The output of the second command shows the full data for the gold table:

external_id	first_name	last_name	dob	email	phone	partner_code
1234567890A	John	Doe	1955-03-15	john.doe@email.com	555-123-4567	ACME
9876543210B	Jane	Smith	1948-07-22	jane.smith@email.com	555-987-6543	ACME
BC-001	Alice	Johnson	1965-08-10	alice.j@test.com	555-222-3333	BETTERCARE
BC-002	Charlie	Brown	1972-03-25	charlie.b@test.com	555-444-5555	BETTERCARE

At the bottom of the cell, there are instructions: '[Shift+Enter] to run and move to next cell' and '[Cmd+Shift+P] to open the command palette'.