INDIAN DATA CLUB

CODE BASICS

databricks

14 DAYS

AI CHALLENGE

DAY 09

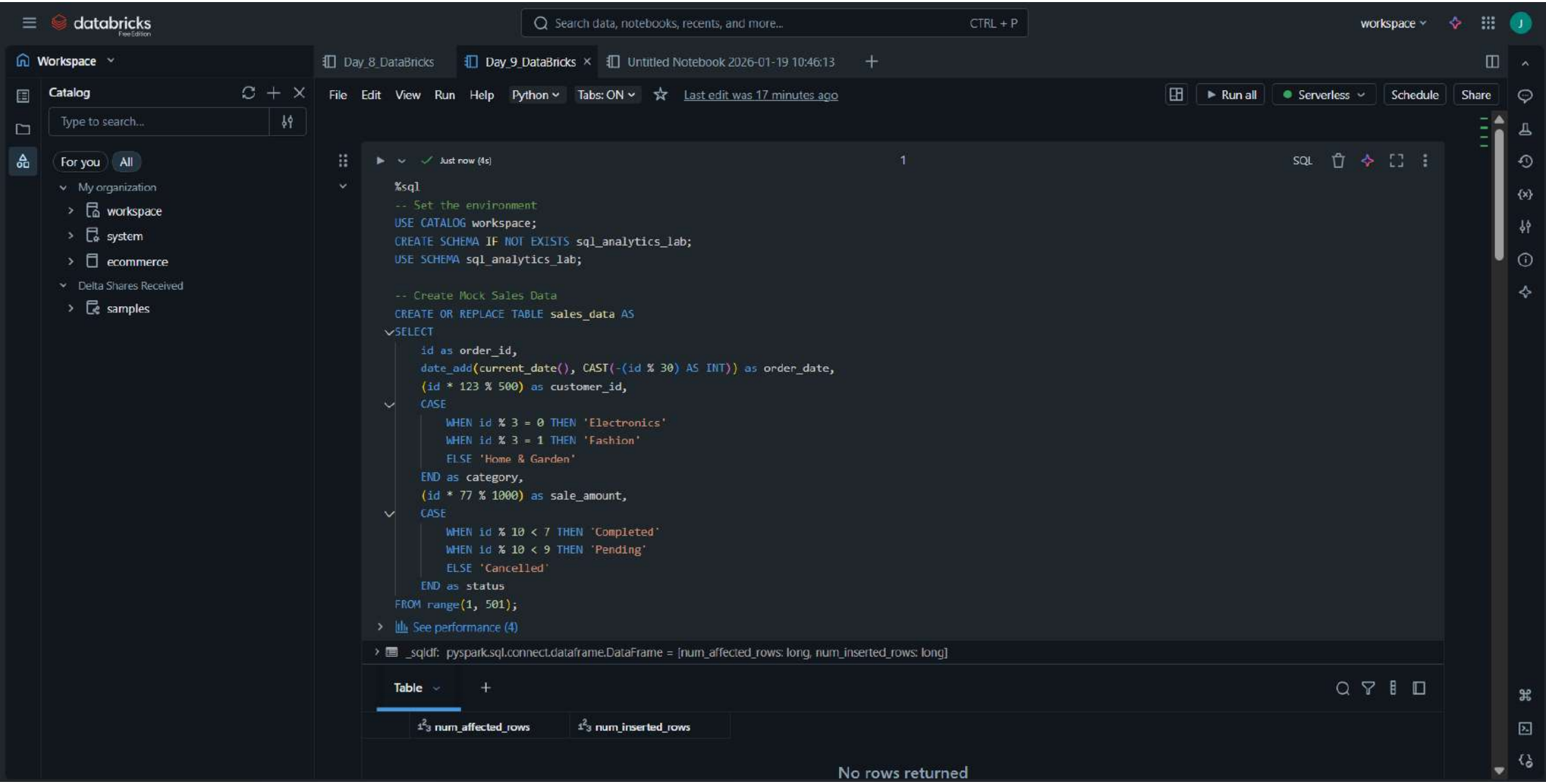
Topic:

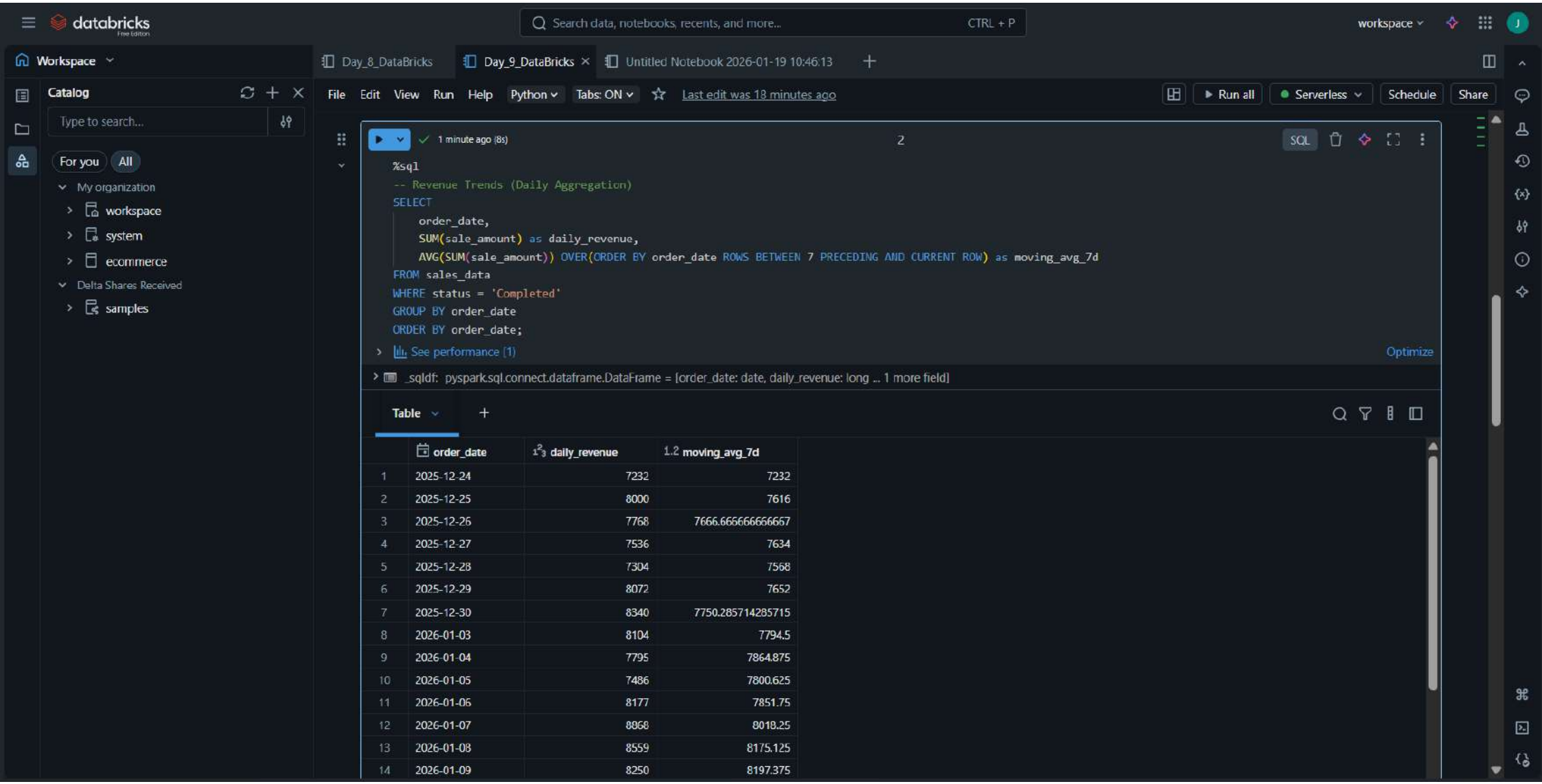
SQL Analytics & Dashboards

Challenge:

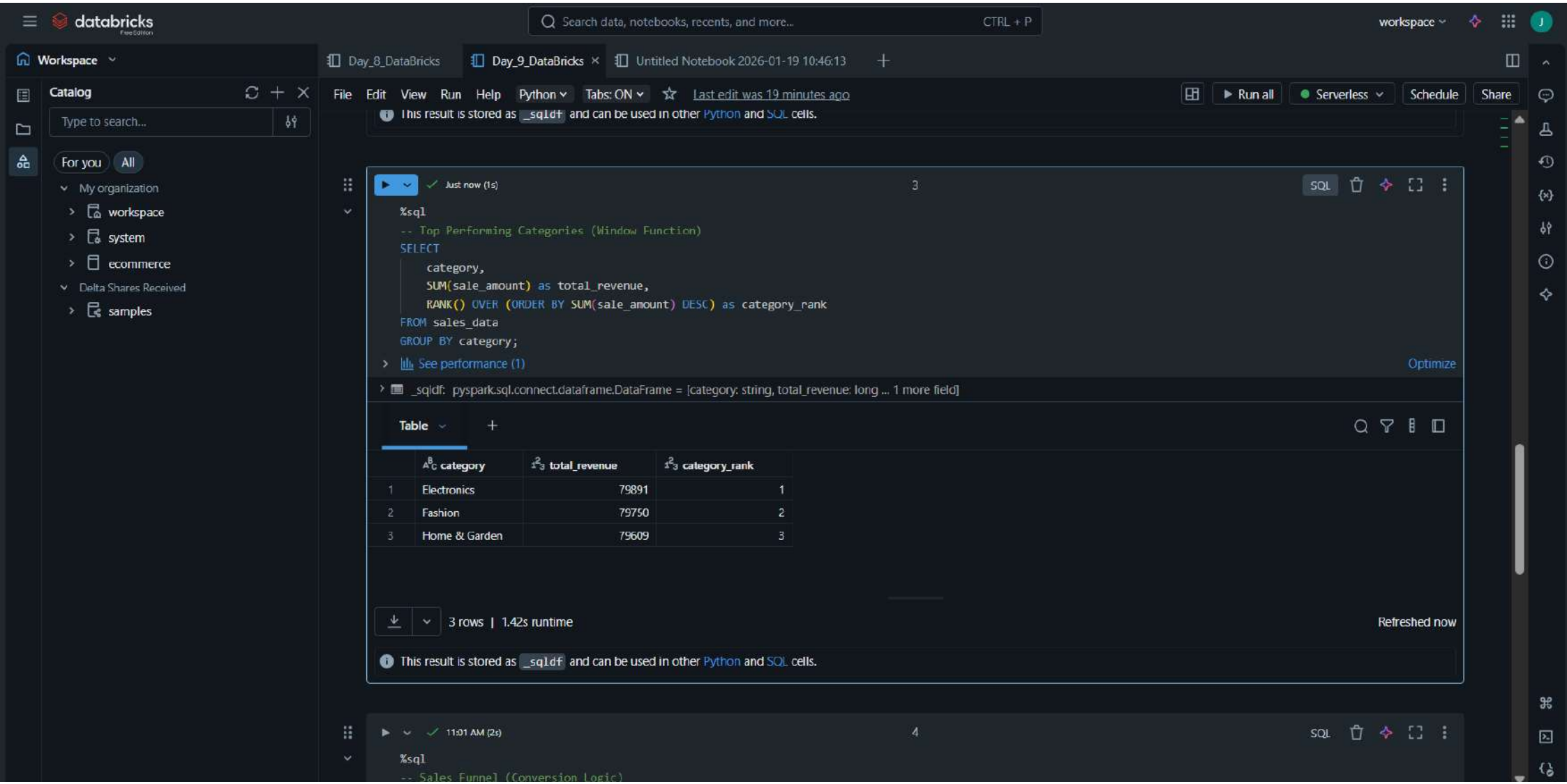
1. Create SQL warehouse
2. Write analytical queries
3. Build dashboard
4. Add filters & schedule refresh

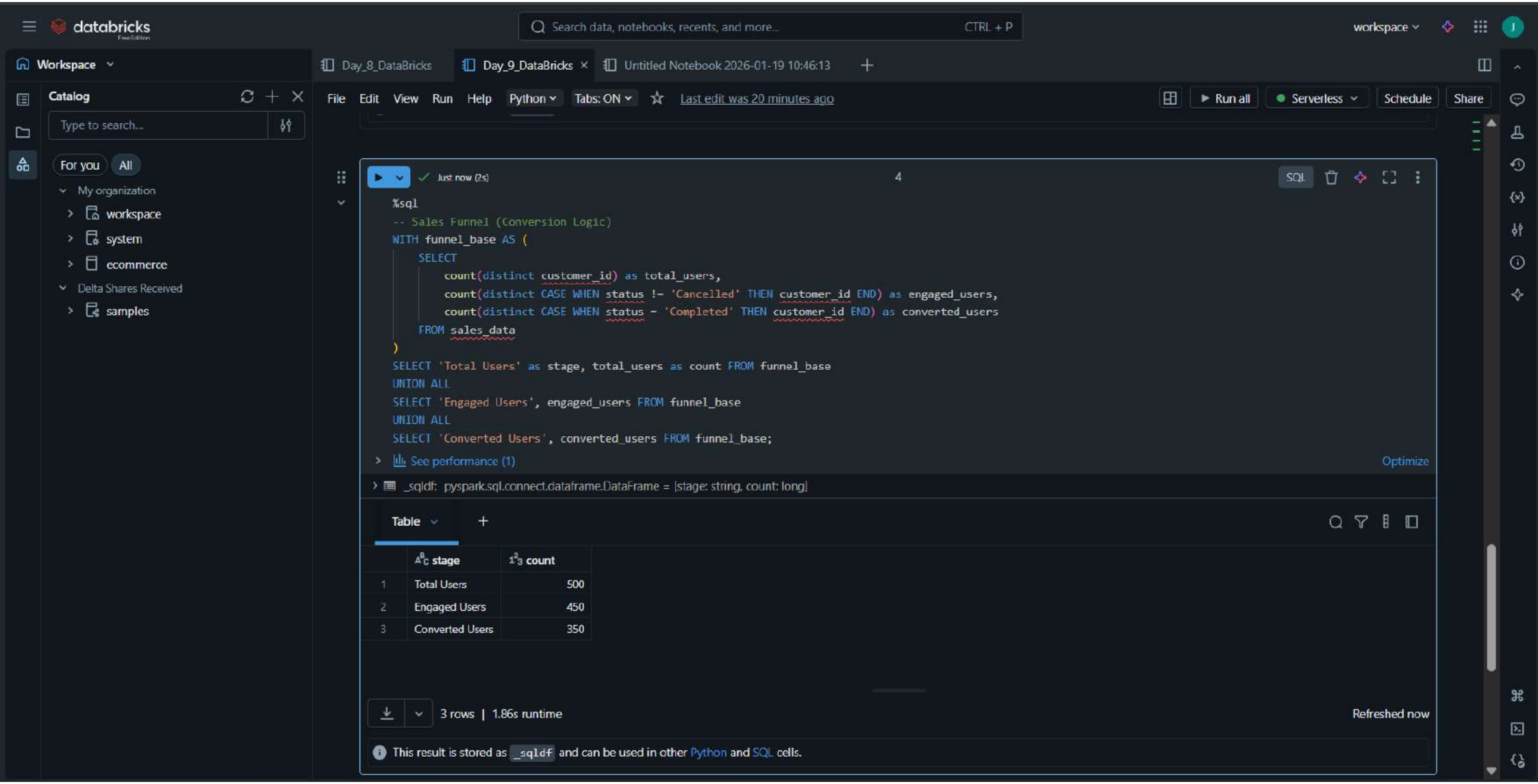
#DatabricksWithIDC

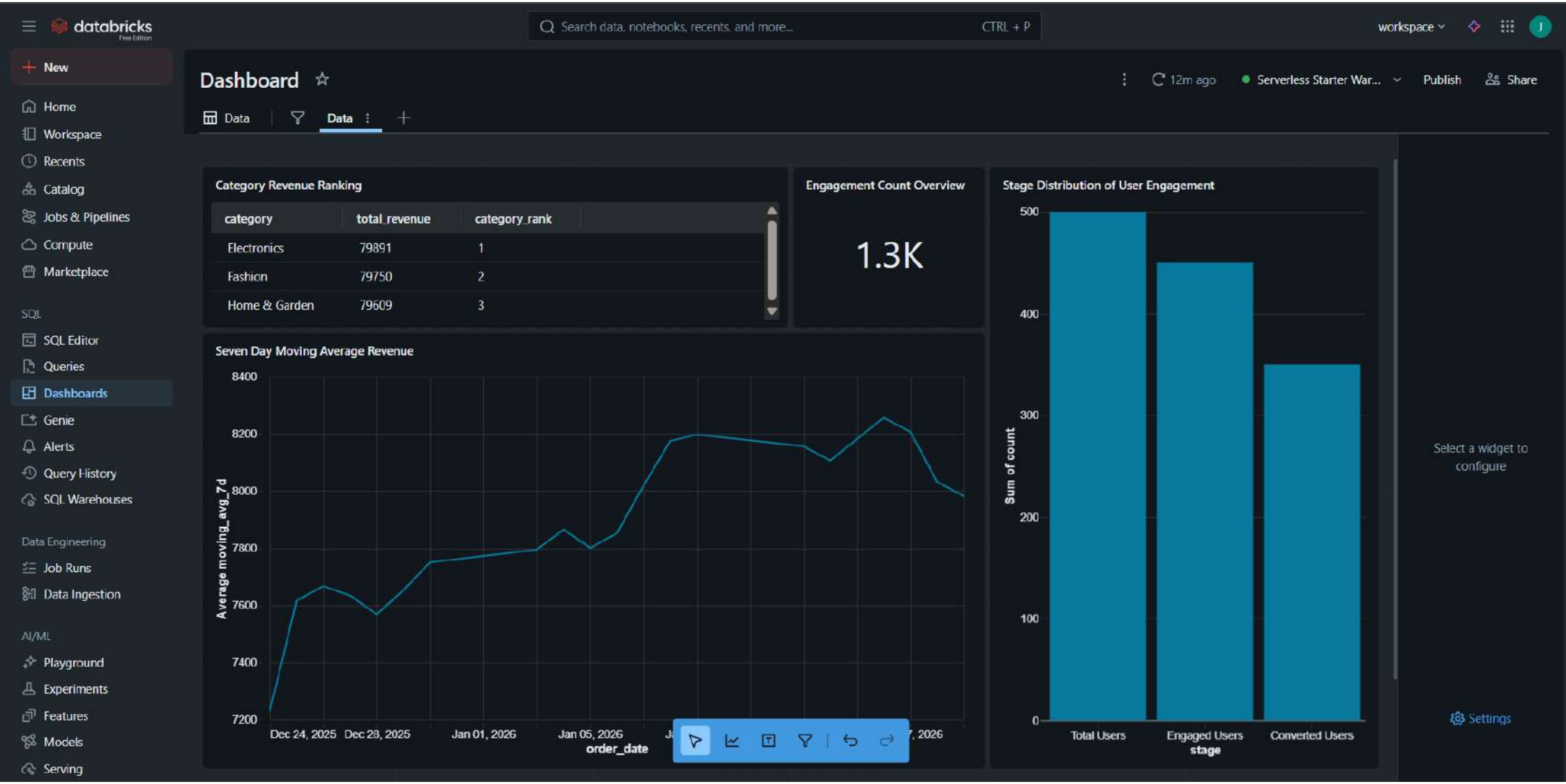

databricks
Free Edition
Search data, notebooks, recents, and more...
CTRL + P
workspace
Workspace
Day_8_DataBricks
Day_9_DataBricks
Untitled Notebook 2026-01-19 10:46:13
Catalog
Type to search...
For you
All
My organization
workspace
system
ecommerce
Delta Shares Received
samples
File
Edit
View
Run
Help
Python
Tabs: ON
Last edit was 17 minutes ago
Run all
Serverless
Schedule
Share
Just now (4s)
1
SQL
%sql
-- Set the environment
USE CATALOG workspace;
CREATE SCHEMA IF NOT EXISTS sql_analytics_lab;
USE SCHEMA sql_analytics_lab;
-- Create Mock Sales Data
CREATE OR REPLACE TABLE sales_data AS
SELECT
id as order_id,
date_add(current_date(), CAST(-(id % 30) AS INT)) as order_date,
(id * 123 % 500) as customer_id,
CASE
WHEN id % 3 = 0 THEN 'Electronics'
WHEN id % 3 = 1 THEN 'Fashion'
ELSE 'Home & Garden'
END as category,
(id * 77 % 1000) as sale_amount,
CASE
WHEN id % 10 < 7 THEN 'Completed'
WHEN id % 10 < 9 THEN 'Pending'
ELSE 'Cancelled'
END as status
FROM range(1, 501);
See performance (4)
_sqldf: pyspark.sql.connect.dataframe.DataFrame = [num_affected_rows: long, num_inserted_rows: long]
Table
num_affected_rows
num_inserted_rows
No rows returned



	order_date	daily_revenue	moving_avg_7d
1	2025-12-24	7232	7232
2	2025-12-25	8000	7616
3	2025-12-26	7768	7666.666666666667
4	2025-12-27	7536	7634
5	2025-12-28	7304	7568
6	2025-12-29	8072	7652
7	2025-12-30	8340	7750.285714285715
8	2026-01-03	8104	7794.5
9	2026-01-04	7795	7864.875
10	2026-01-05	7486	7800.625
11	2026-01-06	8177	7851.75
12	2026-01-07	8868	8018.25
13	2026-01-08	8559	8175.125
14	2026-01-09	8250	8197.375


databricks
Search data, notebooks, recents, and more...
CTRL + P
workspace
Workspace
Day_8_DataBricks
Day_9_DataBricks
Untitled Notebook 2026-01-19 10:46:13
Catalog
Type to search...
For you
All
My organization
workspace
system
ecommerce
Delta Shares Received
samples
File Edit View Run Help Python Tabs: ON
Last edit was 19 minutes ago
Run all
Serverless
Schedule
Share
This result is stored as _sqldf and can be used in other Python and SQL cells.
Just now (1s)
3
SQL
%sql
-- Top Performing Categories (Window Function)
SELECT
category,
SUM(sale_amount) as total_revenue,
RANK() OVER (ORDER BY SUM(sale_amount) DESC) as category_rank
FROM sales_data
GROUP BY category;
See performance (1)
Optimize
_sqldf: pyspark.sql.connect.dataframe.DataFrame = [category: string, total_revenue: long ... 1 more field]
Table
category
total_revenue
category_rank
1 Electronics 79891 1
2 Fashion 79750 2
3 Home & Garden 79609 3
3 rows | 1.42s runtime
Refreshed now
This result is stored as _sqldf and can be used in other Python and SQL cells.
11:01 AM (2s)
4
SQL
%sql
-- Sales Funnel (Conversion Logic)


databricks
Search data, notebooks, recents, and more... CTRL + P
workspace
Workspace
Day_8_DataBricks
Day_9_DataBricks
Untitled Notebook 2026-01-19 10:46:13
Catalog
Type to search...
For you
All
My organization
workspace
system
ecommerce
Delta Shares Received
samples
File Edit View Run Help Python Tabs: ON
Last edit was 20 minutes ago
Run all
Serverless
Schedule
Share
Just now (2s)
4
SQL
%sql
-- Sales Funnel (Conversion Logic)
WITH funnel_base AS (
SELECT
count(distinct customer_id) as total_users,
count(distinct CASE WHEN status != 'Cancelled' THEN customer_id END) as engaged_users,
count(distinct CASE WHEN status = 'Completed' THEN customer_id END) as converted_users
FROM sales_data
)
SELECT 'Total Users' as stage, total_users as count FROM funnel_base
UNION ALL
SELECT 'Engaged Users', engaged_users FROM funnel_base
UNION ALL
SELECT 'Converted Users', converted_users FROM funnel_base;
See performance (1)
Optimize
_sqldf: pyspark.sql.connect.dataframe.DataFrame = [stage: string, count: long]
Table
stage
count
1 Total Users 500
2 Engaged Users 450
3 Converted Users 350
3 rows | 1.86s runtime
Refreshed now
This result is stored as _sqldf and can be used in other Python and SQL cells.


databricks
Free Edition
Search data, notebooks, recents, and more...
CTRL + P
workspace
New
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
Features
Models
Serving
Dashboard
12m ago
Serverless Starter War...
Publish
Share
Data
Data
Category Revenue Ranking
category
total_revenue
category_rank
Electronics
79891
1
Fashion
79750
2
Home & Garden
79609
3
Engagement Count Overview
1.3K
Stage Distribution of User Engagement
Sum of count
500
400
300
200
100
0
Total Users
Engaged Users
Converted Users
stage
Seven Day Moving Average Revenue
Average moving_avg_7d
8400
8200
8000
7800
7600
7400
7200
Dec 24, 2025
Dec 28, 2025
Jan 01, 2026
Jan 05, 2026
order_date
Select a widget to configure
Settings