INDIAN DATA CLUB

CODE BASICS

databricks

14 DAYS

AI CHALLENGE

DAY 10

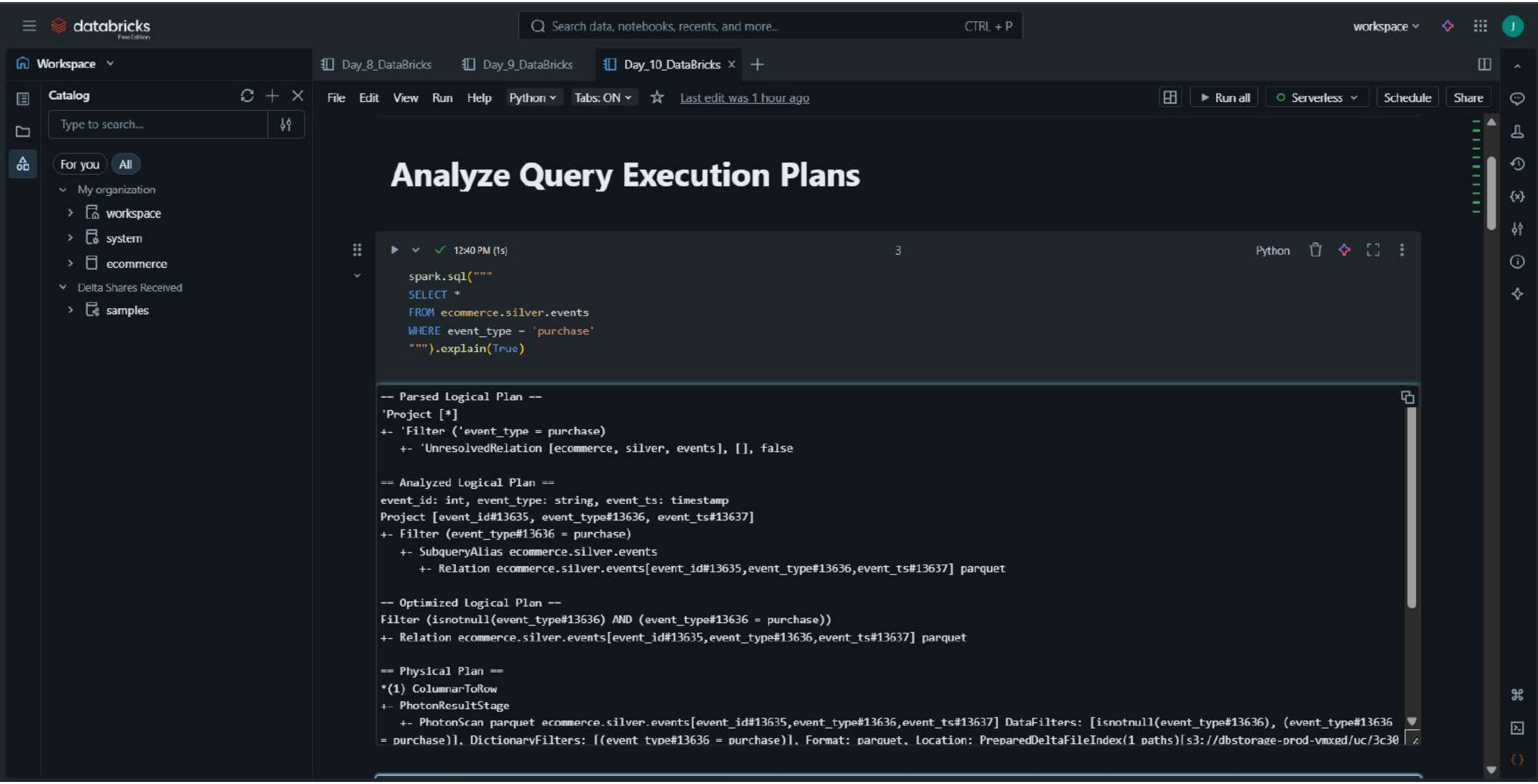
Topic:

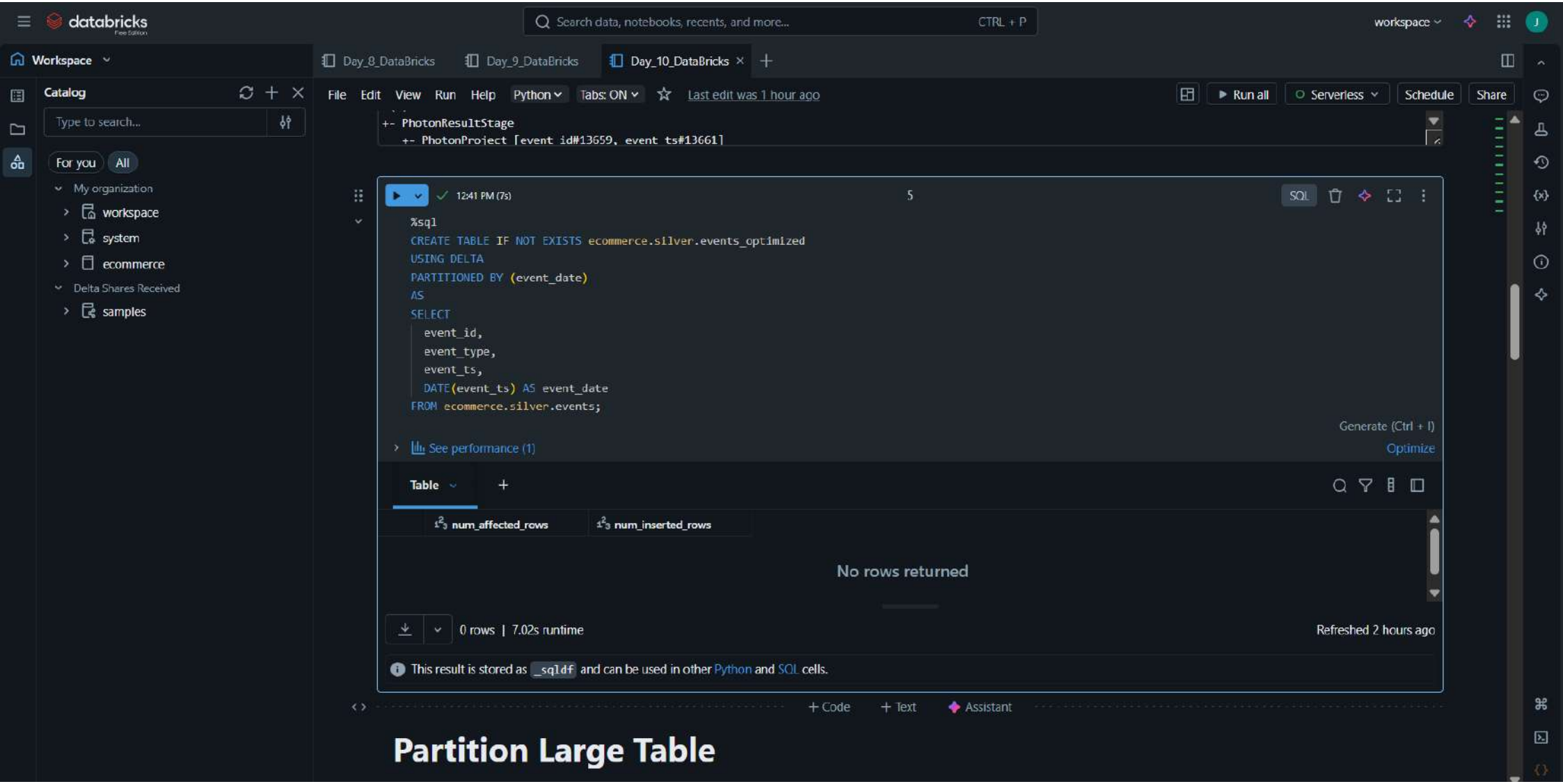
Performance Optimization

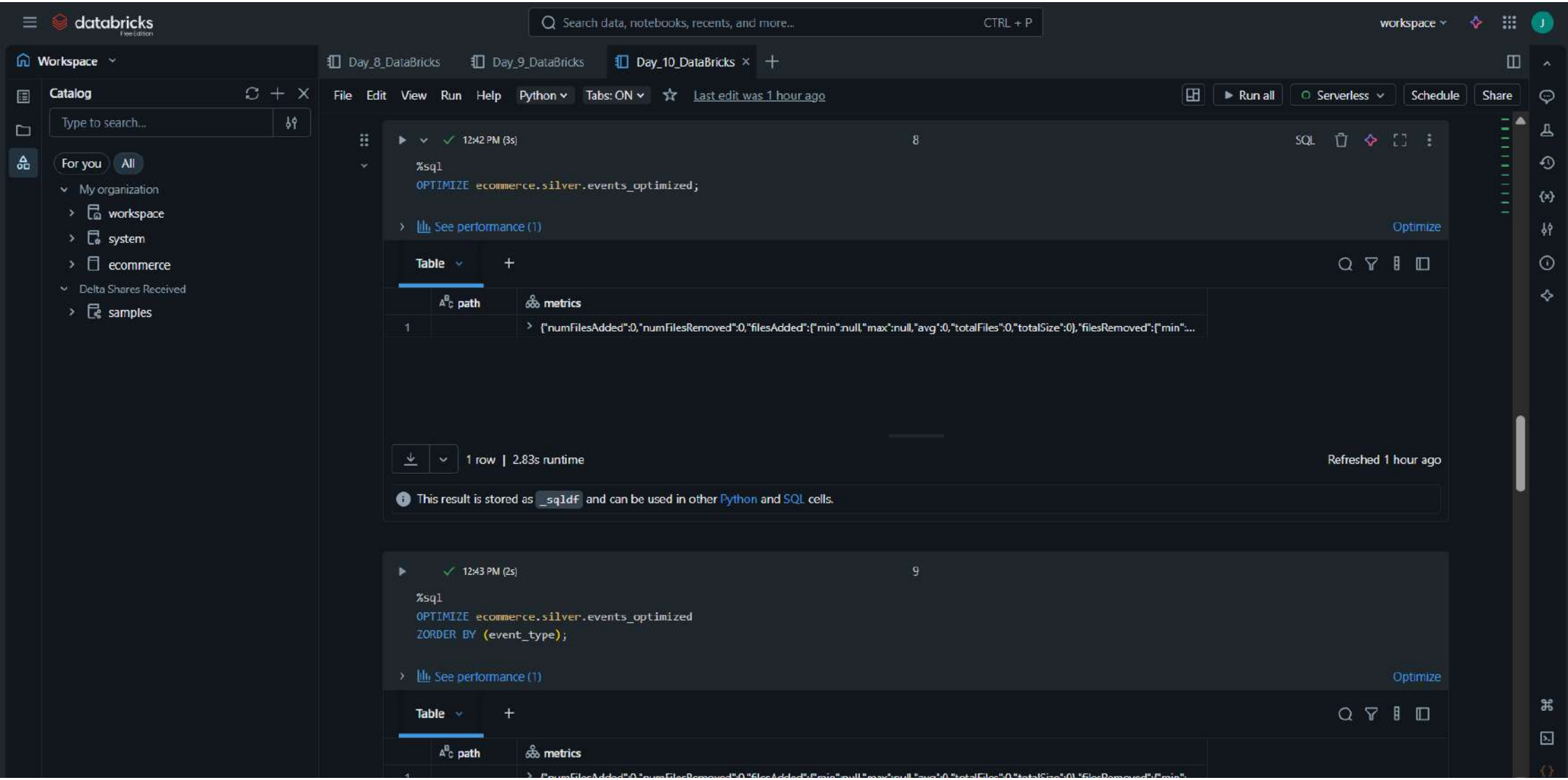
Challenge:

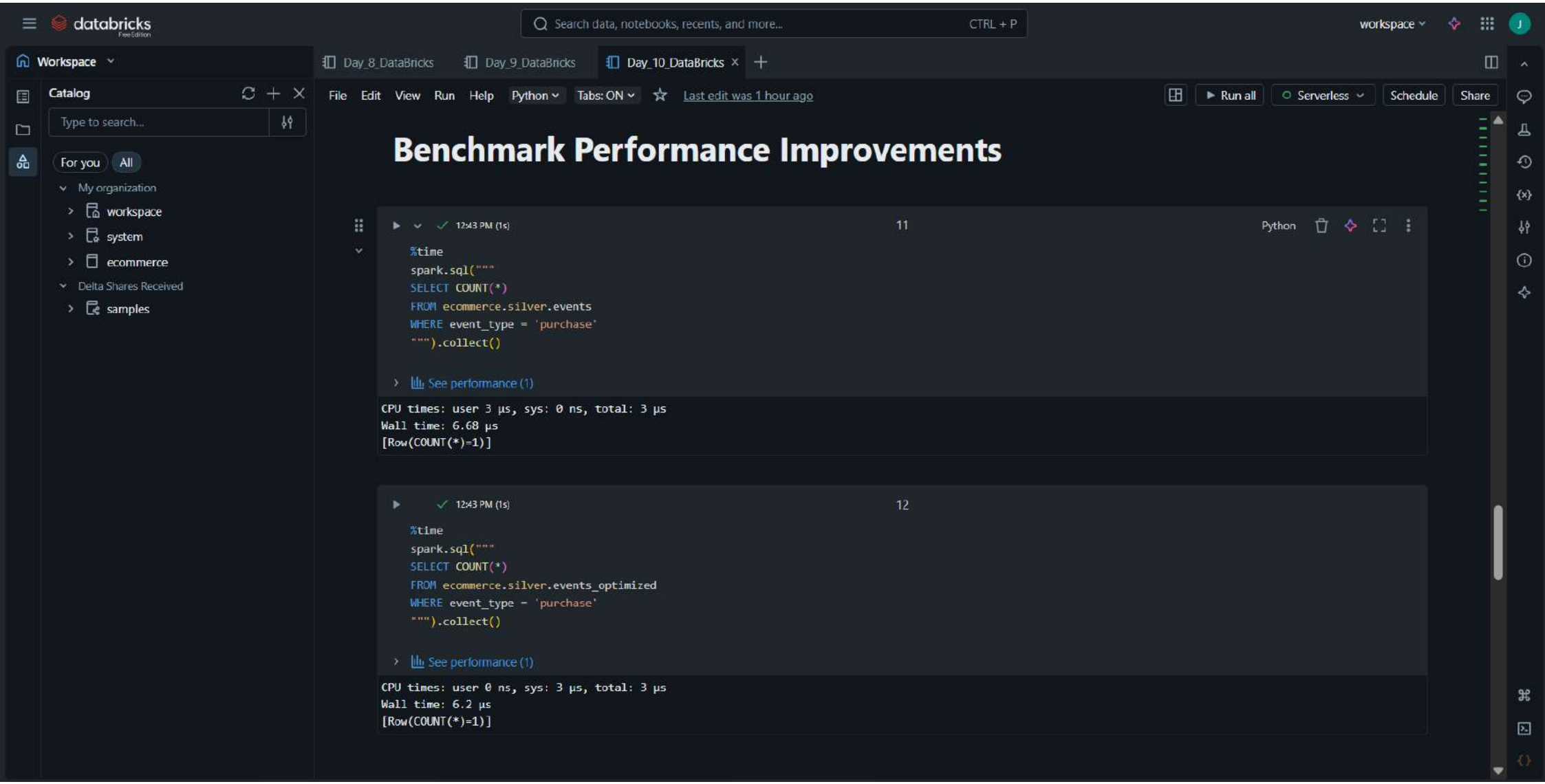
1. Analyze query plans
2. Partition large tables
3. Apply ZORDER
4. Benchmark improvements

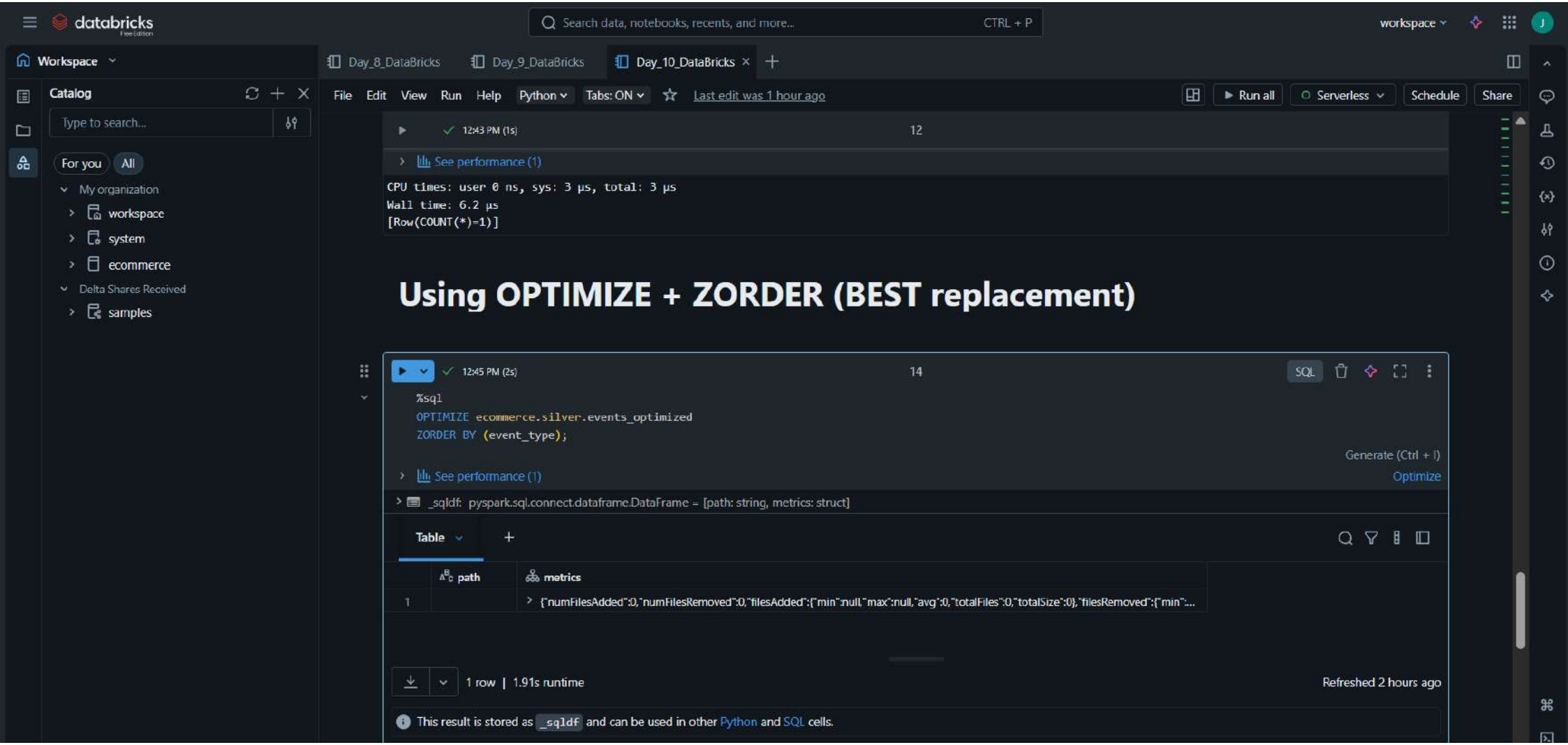
#DatabricksWithIDC

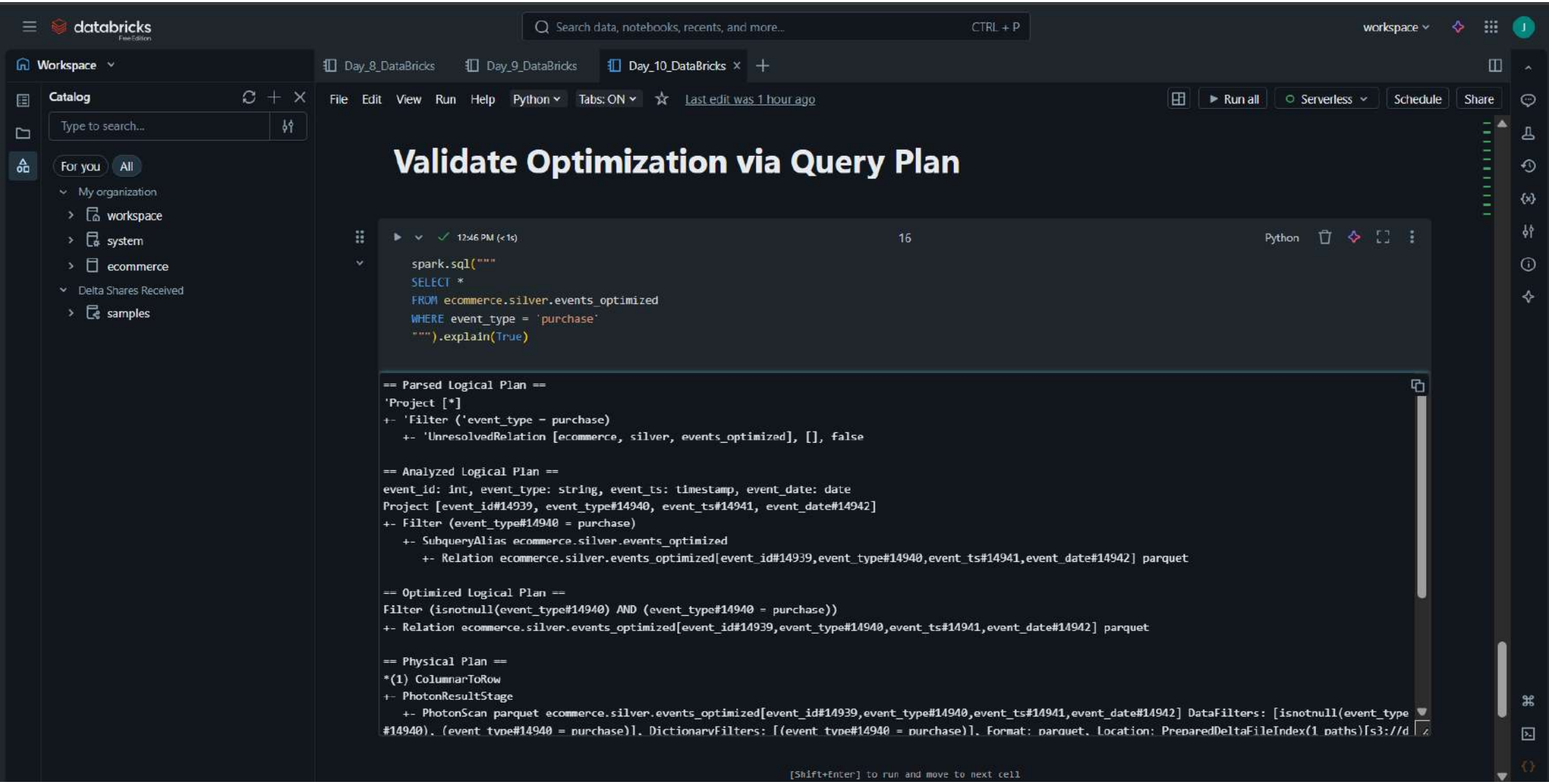

databricks
Search data, notebooks, recents, and more...
CTRL + P
workspace
Workspace
Day_8_DataBricks
Day_9_DataBricks
Day_10_DataBricks
Catalog
Type to search...
For you
All
My organization
workspace
system
ecommerce
Delta Shares Received
samples
File Edit View Run Help Python Tabs: ON
Last edit was 1 hour ago
Run all
Serverless
Schedule
Share
Analyze Query Execution Plans
12:40 PM (1s)
3
Python
spark.sql("""
SELECT *
FROM ecommerce.silver.events
WHERE event_type = 'purchase'
""").explain(True)
-- Parsed Logical Plan --
'Project [*]
+- 'Filter ('event_type = purchase)
+- 'UnresolvedRelation [ecommerce, silver, events], [], false
== Analyzed Logical Plan ==
event_id: int, event_type: string, event_ts: timestamp
Project [event_id#13635, event_type#13636, event_ts#13637]
+- Filter (event_type#13636 = purchase)
+- SubqueryAlias ecommerce.silver.events
+- Relation ecommerce.silver.events[event_id#13635,event_type#13636,event_ts#13637] parquet
-- Optimized Logical Plan --
Filter (isnotnull(event_type#13636) AND (event_type#13636 = purchase))
+- Relation ecommerce.silver.events[event_id#13635,event_type#13636,event_ts#13637] parquet
== Physical Plan ==
*(1) ColumnarToRow
+- PhotonResultStage
+- PhotonScan parquet ecommerce.silver.events[event_id#13635,event_type#13636,event_ts#13637] DataFilters: [isnotnull(event_type#13636), (event_type#13636
= purchase)], DictionaryFilters: [(event_type#13636 = purchase)], Format: parquet, Location: PreparedDeltaFileIndex(1 paths)[s3://dbstorage-prod-vmxgd/uc/3c30


databricks
Search data, notebooks, recents, and more... CTRL + P
workspace
Workspace
Day_8_DataBricks
Day_9_DataBricks
Day_10_DataBricks
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 1 hour ago
Run all
Serverless
Schedule
Share
+- PhotonResultStage
+- PhotonProject [event_id#13659, event_ts#13661]
12:41 PM (7s)
5
SQL
%sql
CREATE TABLE IF NOT EXISTS ecommerce.silver.events_optimized
USING DELTA
PARTITIONED BY (event_date)
AS
SELECT
event_id,
event_type,
event_ts,
DATE(event_ts) AS event_date
FROM ecommerce.silver.events;
Generate (Ctrl + I)
See performance (1)
Optimize
Table
num_affected_rows
num_inserted_rows
No rows returned
0 rows | 7.02s runtime
Refreshed 2 hours ago
This result is stored as _sqldf and can be used in other Python and SQL cells.
+ Code
+ Text
Assistant
Partition Large Table


databricks
Free Edition
Search data, notebooks, recents, and more...
CTRL + P
workspace
Workspace
Day_8_DataBricks
Day_9_DataBricks
Day_10_DataBricks
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 1 hour ago
Run all
Serverless
Schedule
Share
12:42 PM (3s)
8
SQL
%sql
OPTIMIZE ecommerce.silver.events_optimized;
See performance (1)
Optimize
Table
path
metrics
1
{"numFilesAdded":0,"numFilesRemoved":0,"filesAdded":{"min":null,"max":null,"avg":0,"totalFiles":0,"totalSize":0},"filesRemoved":{"min":...
1 row | 2.83s runtime
Refreshed 1 hour ago
This result is stored as _sqldf and can be used in other Python and SQL cells.
12:43 PM (2s)
9
%sql
OPTIMIZE ecommerce.silver.events_optimized
ZORDER BY (event_type);
See performance (1)
Optimize
Table
path
metrics


databricks
Free Edition
Search data, notebooks, recents, and more...
CTRL + P
workspace
Workspace
Day_8_DataBricks
Day_9_DataBricks
Day_10_DataBricks
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 1 hour ago
Run all
Serverless
Schedule
Share
Benchmark Performance Improvements
12:43 PM (1s)
11
Python
%time
spark.sql("""
SELECT COUNT(*)
FROM ecommerce.silver.events
WHERE event_type = 'purchase'
""").collect()
See performance (1)
CPU times: user 3 µs, sys: 0 ns, total: 3 µs
Wall time: 6.68 µs
[Row(COUNT(*)=1)]
12:43 PM (1s)
12
%time
spark.sql("""
SELECT COUNT(*)
FROM ecommerce.silver.events_optimized
WHERE event_type = 'purchase'
""").collect()
See performance (1)
CPU times: user 0 ns, sys: 3 µs, total: 3 µs
Wall time: 6.2 µs
[Row(COUNT(*)=1)]


databricks
Search data, notebooks, recents, and more... CTRL + P
workspace
Workspace
Day_8_DataBricks
Day_9_DataBricks
Day_10_DataBricks
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 1 hour ago
Run all
Serverless
Schedule
Share
12:43 PM (1s)
12
See performance (1)
CPU times: user 0 ns, sys: 3 µs, total: 3 µs
Wall time: 6.2 µs
[Row(COUNT(*)=1)]
Using OPTIMIZE + ZORDER (BEST replacement)
12:45 PM (2s)
14
SQL
%sql
OPTIMIZE ecommerce.silver.events_optimized
ZORDER BY (event_type);
Generate (Ctrl + I)
See performance (1)
Optimize
_sqldf: pyspark.sql.connect.dataframe.DataFrame = [path: string, metrics: struct]
Table
path
metrics
1
{"numFilesAdded":0,"numFilesRemoved":0,"filesAdded":{"min":null,"max":null,"avg":0,"totalFiles":0,"totalSize":0},"filesRemoved":{"min":...
1 row | 1.91s runtime
Refreshed 2 hours ago
This result is stored as _sqldf and can be used in other Python and SQL cells.


databricks
Search data, notebooks, recents, and more...
CTRL + P
workspace
Workspace
Day_8_DataBricks
Day_9_DataBricks
Day_10_DataBricks
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 1 hour ago
Run all
Serverless
Schedule
Share
Validate Optimization via Query Plan
12:46 PM (<1s)
16
Python
spark.sql("""
SELECT *
FROM ecommerce.silver.events_optimized
WHERE event_type = 'purchase'
""").explain(True)
== Parsed Logical Plan ==
'Project [*]
+- 'Filter ('event_type = purchase)
+- 'UnresolvedRelation [ecommerce, silver, events_optimized], [], false
== Analyzed Logical Plan ==
event_id: int, event_type: string, event_ts: timestamp, event_date: date
Project [event_id#14939, event_type#14940, event_ts#14941, event_date#14942]
+- Filter (event_type#14940 = purchase)
+- SubqueryAlias ecommerce.silver.events_optimized
+- Relation ecommerce.silver.events_optimized[event_id#14939,event_type#14940,event_ts#14941,event_date#14942] parquet
== Optimized Logical Plan ==
Filter (isnotnull(event_type#14940) AND (event_type#14940 = purchase))
+- Relation ecommerce.silver.events_optimized[event_id#14939,event_type#14940,event_ts#14941,event_date#14942] parquet
== Physical Plan ==
*(1) ColumnarToRow
+- PhotonResultStage
+- PhotonScan parquet ecommerce.silver.events_optimized[event_id#14939,event_type#14940,event_ts#14941,event_date#14942] DataFilters: [isnotnull(event_type
#14940), (event_type#14940 = purchase)], DictionaryFilters: [(event_type#14940 = purchase)], Format: parquet, Location: PreparedDeltaFileIndex(1 paths)[s3://d
[Shift+Enter] to run and move to next cell