



14 DAYS

AI CHALLENGE

DAY 10

Topic:

Performance Optimization

Challenge:

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

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace

Day_8_DataBrics Day_9_DataBrics Day_10_DataBrics +

Workspace Catalog

Type to search...

For you All

- My organization
 - workspace
 - system
 - ecommerce
- Delta Shares Received
- samples

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-vmxed/uc/3c30]

Databricks Notebook interface showing a SQL cell running a query to create a partitioned table.

Catalog sidebar:

- Type to search...
- For you
- All
- My organization
 - workspace
 - system
 - ecommerce
- Delta Shares Received
- samples

Search bar: Search data, notebooks, recents, and more... CTRL + P

Tab bar: Day_8_DataBrics, Day_9_DataBrics, Day_10_DataBrics (active), +

Toolbar: File, Edit, View, Run, Help, Python, Tabs: ON, Last edit was 1 hour ago, Run all, Serverless, Schedule, Share

Code Cell:

```
%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;
```

Execution status: 12:41 PM (7s)

Performance metrics: 5 rows, 0 affected, 0 inserted, 0 updated, 0 deleted, 0 errors, 7.02s runtime.

No rows returned.

This result is stored as `_sqldf` and can be used in other Python and SQL cells.

Bottom navigation: + Code, + Text, Assistant

Section Header: Partition Large Table

dataBricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace ▾

Day_8_DataBricks Day_9_DataBricks Day_10_DataBricks +

Workspace Catalog

Type to search...

For you All

- My organization
 - workspace
 - system
 - ecommerce
- DeltaShares Received
- samples

File Edit View Run Help Python Tabs: ON Last edit was 1 hour ago

12:42 PM (3s) 8

```
%sql  
OPTIMIZE ecommerce.silver.events_optimized;
```

See performance (1) Optimize

Table +

A ^B c 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 +

A ^B c path	metrics
1	> {"numFilesAdded":0,"numFilesRemoved":0,"filesAdded":{"min":null,"max":null,"avg":0,"totalFiles":0,"totalSize":0}, "filesRemoved":{"min":...

Databricks Free Edition

Search data, notebooks, recents, and more... CTRL + P

workspace ▾ J

Workspace Catalog

Type to search... 🔍

For you All

- My organization
 - workspace
 - system
 - ecommerce
- Delta Shares Received
- samples

Day_8_DataBrics Day_9_DataBrics Day_10_DataBrics +

File Edit View Run Help Python Tabs: ON Last edit was 1 hour ago

Run all Serverless Schedule Share

Benchmark Performance Improvements

11

```
%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

```
%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)]

This screenshot shows the Databricks workspace interface. On the left is the sidebar with navigation links like 'Workspace' and 'Catalog'. The main area displays two code cells. The first cell, labeled '11', contains a simple SQL query using the `spark.sql` method. The second cell, labeled '12', contains the same query but uses the `events_optimized` table instead. Both cells show execution results with CPU and wall times. A sidebar on the right provides various workspace management options.

Screenshot of the Databricks workspace interface showing a notebook titled "Day_10_DataBrics".

The left sidebar includes:

- Workspace
- Catalog
- Type to search...
- For you
- All
- My organization
- workspace
- system
- ecommerce
- DeltaShares Received
- samples

The main area shows a notebook tab bar with tabs: Day_8_DataBrics, Day_9_DataBrics, Day_10_DataBrics (active), and a new tab button. The top navigation bar includes: File, Edit, View, Run, Help, Python, Tabs: ON, Last edit was 1 hour ago, and workspace dropdown.

The notebook content displays a cell output:

```
▶ 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)]
```

A large title "Using OPTIMIZE + ZORDER (BEST replacement)" is centered above another cell.

The second cell shows the following content:

```
%sql
OPTIMIZE ecommerce.silver.events_optimized
ZORDER BY (event_type);
```

Output of the cell:

```
▶ See performance (1)
> _sqldf: pyspark.sql.connect.DataFrame = [path: string, metrics: struct]
Table 1
path metrics
1 > {"numFilesAdded":0,"numFilesRemoved":0,"filesAdded":{"min":null,"max":null,"avg":0,"totalFiles":0,"totalSize":0}, "filesRemoved":{"min":...}
```

Bottom status bar: 1 row | 1.91s runtime, Refreshed 2 hours ago.

Note: A tooltip at the bottom states: "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 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

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
```