INDIAN DATA CLUB

CODE BASICS

databricks

14 DAYS

AI CHALLENGE

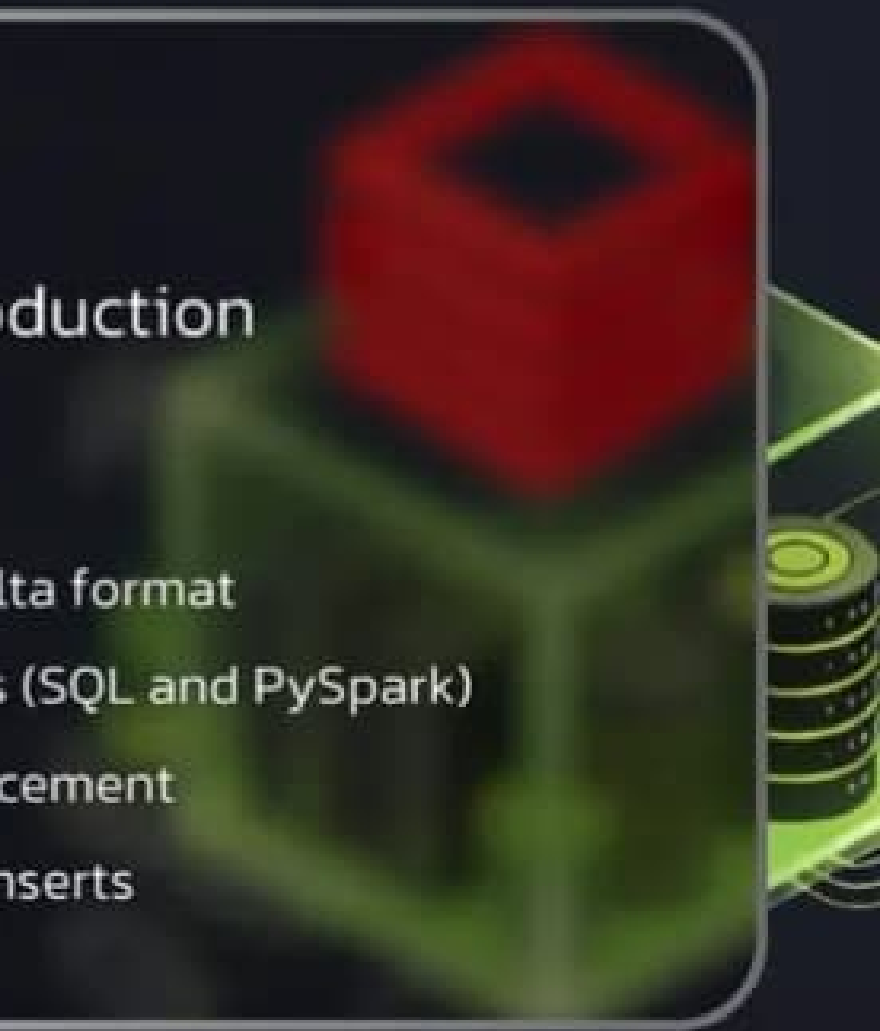
DAY 04

Topic:

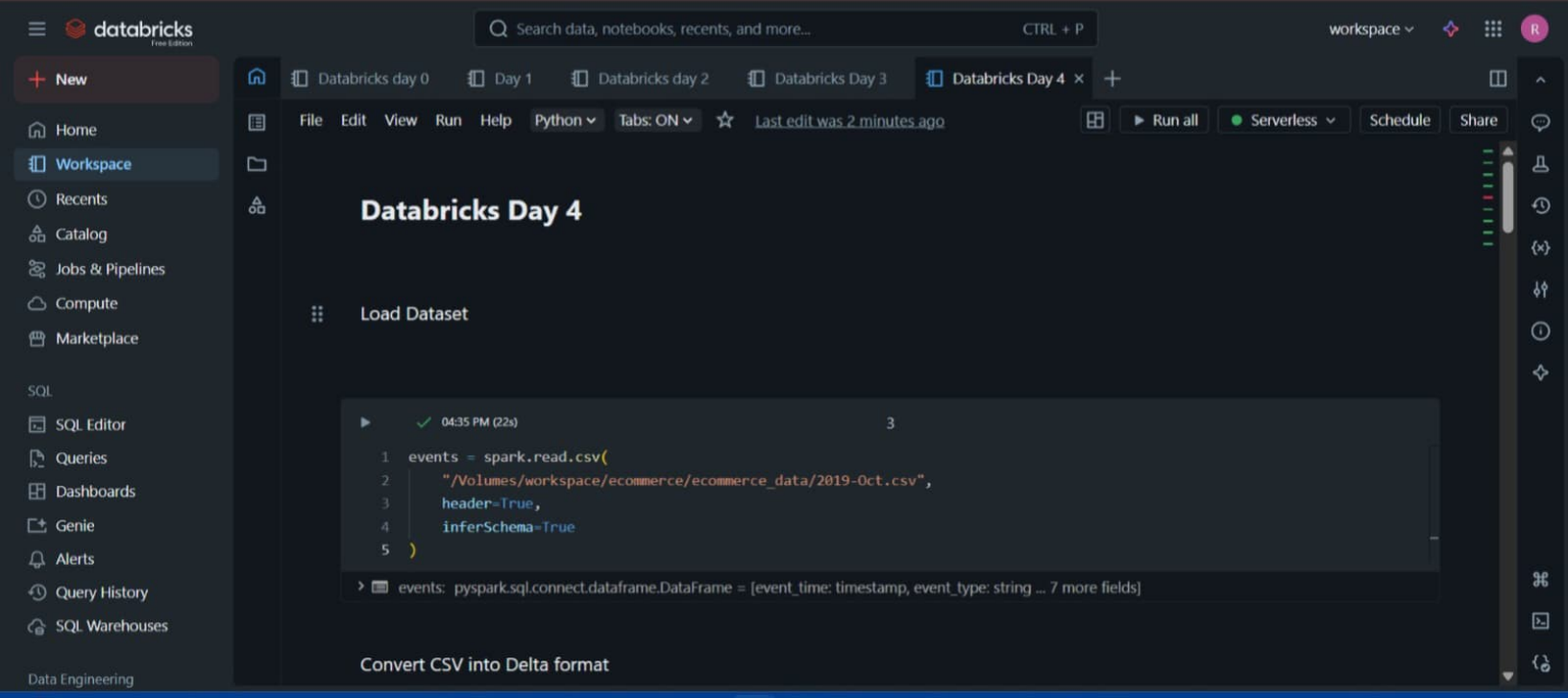
Delta Lake Introduction

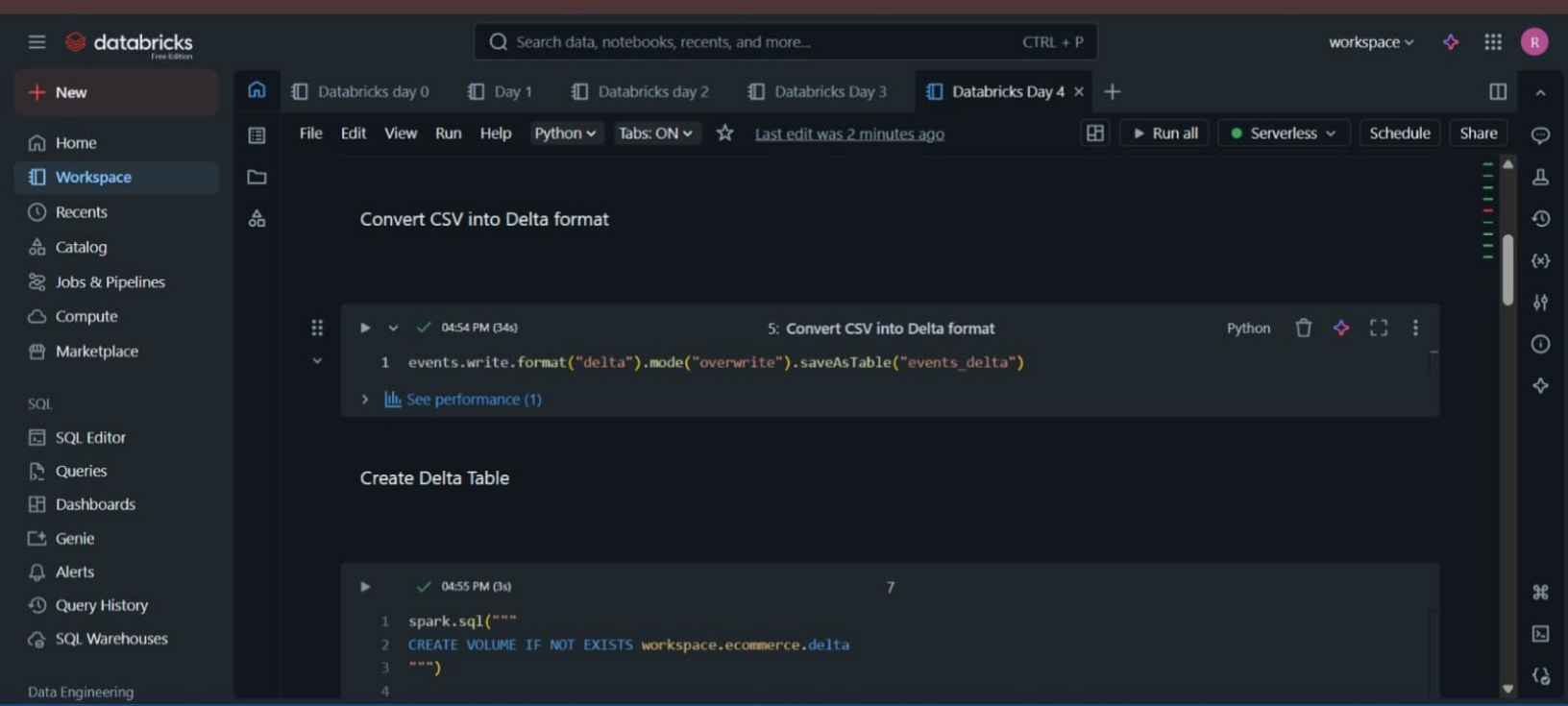
Challenge:

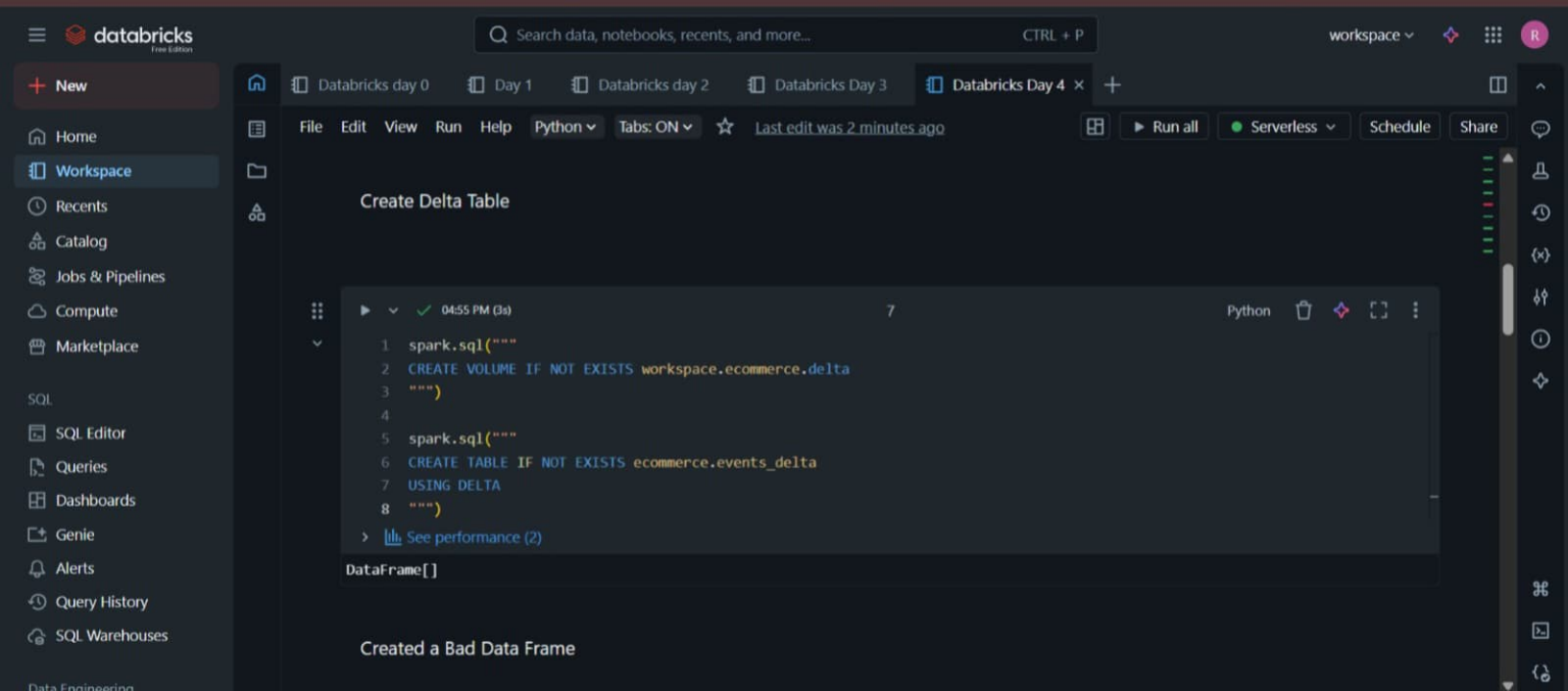
1. Convert CSV to Delta format
2. Create Delta tables (SQL and PySpark)
3. Test schema enforcement
4. Handle duplicate inserts

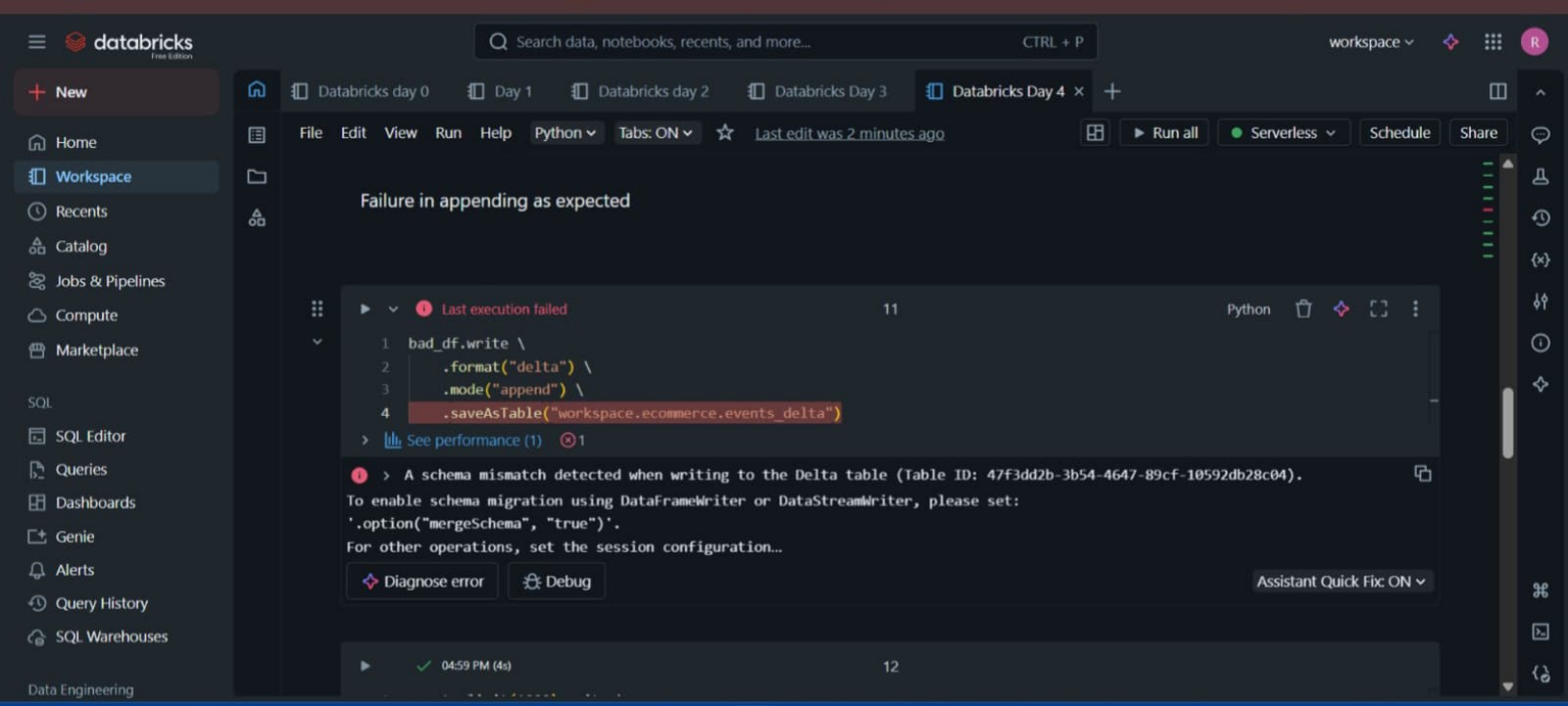


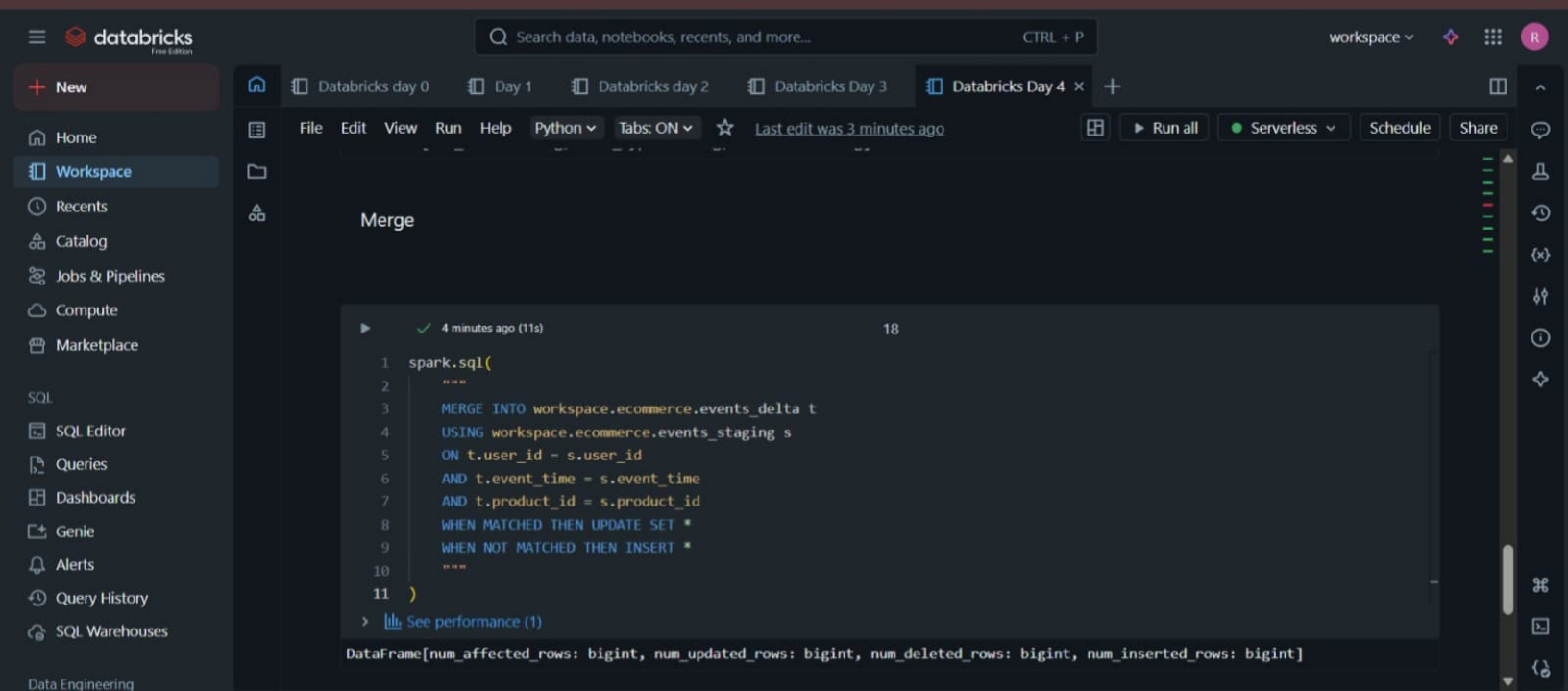
#DatabricksWithIDC


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
Databricks day 0
Day 1
Databricks day 2
Databricks Day 3
Databricks Day 4
File
Edit
View
Run
Help
Python
Tabs: ON
Last edit was 2 minutes ago
Run all
Serverless
Schedule
Share
Databricks Day 4
Load Dataset
04:35 PM (22s)
3
1 events = spark.read.csv(
2 "/Volumes/workspace/ecommerce/ecommerce_data/2019-Oct.csv",
3 header=True,
4 inferSchema=True
5)
events: pyspark.sql.connect.dataframe.DataFrame = [event_time: timestamp, event_type: string ... 7 more fields]
Convert CSV into Delta format


databricks
Free Edition
Search data, notebooks, recents, and more...
CTRL + P
workspace
R
New
Home
Workspace
Recents
Catalog
Jobs & Pipelines
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Databricks day 0
Day 1
Databricks day 2
Databricks Day 3
Databricks Day 4
File
Edit
View
Run
Help
Python
Tabs: ON
Last edit was 2 minutes ago
Run all
Serverless
Schedule
Share
Convert CSV into Delta format
04:54 PM (34s)
5: Convert CSV into Delta format
Python
1 events.write.format("delta").mode("overwrite").saveAsTable("events_delta")
See performance (1)
Create Delta Table
04:55 PM (3s)
7
1 spark.sql("""
2 CREATE VOLUME IF NOT EXISTS workspace.ecommerce.delta
3 """)
4


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
Databricks day 0
Day 1
Databricks day 2
Databricks Day 3
Databricks Day 4
File Edit View Run Help Python Tabs: ON
Last edit was 2 minutes ago
Run all
Serverless
Schedule
Share
Create Delta Table
04:55 PM (3s)
7
Python
spark.sql("""
CREATE VOLUME IF NOT EXISTS workspace.ecommerce.delta
""")
spark.sql("""
CREATE TABLE IF NOT EXISTS ecommerce.events_delta
USING DELTA
""")
See performance (2)
DataFrame[]
Created a Bad Data Frame


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
Databricks day 0
Day 1
Databricks day 2
Databricks Day 3
Databricks Day 4
File Edit View Run Help Python Tabs: ON
Last edit was 2 minutes ago
Run all
Serverless
Schedule
Share
Failure in appending as expected
Last execution failed
11
Python
bad_df.write \
.format("delta") \
.mode("append") \
.saveAsTable("workspace.ecommerce.events_delta")
See performance (1)
1
A schema mismatch detected when writing to the Delta table (Table ID: 47f3dd2b-3b54-4647-89cf-10592db28c04).
To enable schema migration using DataFrameWriter or DataStreamWriter, please set:
'.option("mergeSchema", "true")'.
For other operations, set the session configuration...
Diagnose error
Debug
Assistant Quick Fix: ON
04:59 PM (4s)
12


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
Databricks day 0
Day 1
Databricks day 2
Databricks Day 3
Databricks Day 4
File Edit View Run Help Python Tabs: ON
Last edit was 3 minutes ago
Run all
Serverless
Schedule
Share
Merge
4 minutes ago (11s)
18
spark.sql(
"""
MERGE INTO workspace.ecommerce.events_delta t
USING workspace.ecommerce.events_staging s
ON t.user_id = s.user_id
AND t.event_time = s.event_time
AND t.product_id = s.product_id
WHEN MATCHED THEN UPDATE SET *
WHEN NOT MATCHED THEN INSERT *
"""
)
See performance (1)
DataFrame[num_affected_rows: bigint, num_updated_rows: bigint, num_deleted_rows: bigint, num_inserted_rows: bigint]