INDIAN DATA CLUB

CODE BASICS

databricks

14 DAYS

AI CHALLENGE

DAY 03

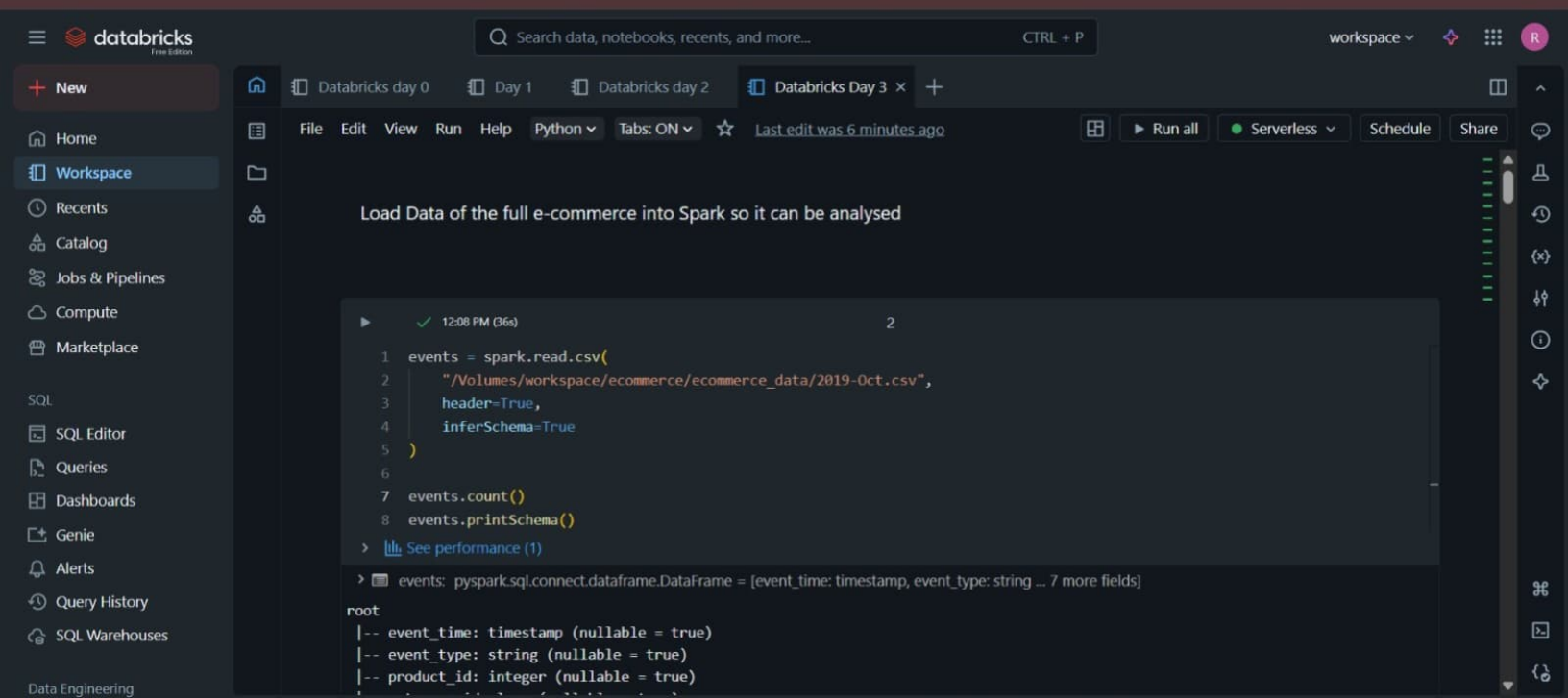
Topic:

PySpark Transformations Deep Dive

Challenge:

1. Load full e-commerce dataset
2. Perform complex joins
3. Calculate running totals with window functions
4. Create derived features

#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
File Edit View Run Help Python Tabs: ON
Last edit was 6 minutes ago
Run all
Serverless
Schedule
Share
Load Data of the full e-commerce into Spark so it can be analysed
12:08 PM (36s)
2
events = spark.read.csv(
"/Volumes/workspace/ecommerce/ecommerce_data/2019-Oct.csv",
header=True,
inferSchema=True
)
events.count()
events.printSchema()
See performance (1)
events: pyspark.sql.connect.dataframe.DataFrame = [event_time: timestamp, event_type: string ... 7 more fields]
root
|-- event_time: timestamp (nullable = true)
|-- event_type: string (nullable = true)
|-- product_id: integer (nullable = true)

New
Home
Workspace
Recents
Catalog
Jobs & Pipelines
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering

Track Customer Activity Step-by-Step

10:43 AM (<1s) 15

```
1 from pyspark.sql.window import Window
2 user_window = Window.partitionBy("user_id").orderBy("event_time")
```

created running activity count

10:46 AM (16s) 17

```
1 user_activity = events.withColumn("cumulative_events",
2                                   F.count("*").over(user_window))
3
4 user_activity.select("user_id","event_time","cumulative_events").show(15)
```

New
Home
Workspace
Recents
Catalog
Jobs & Pipelines
Compute
Marketplace
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering

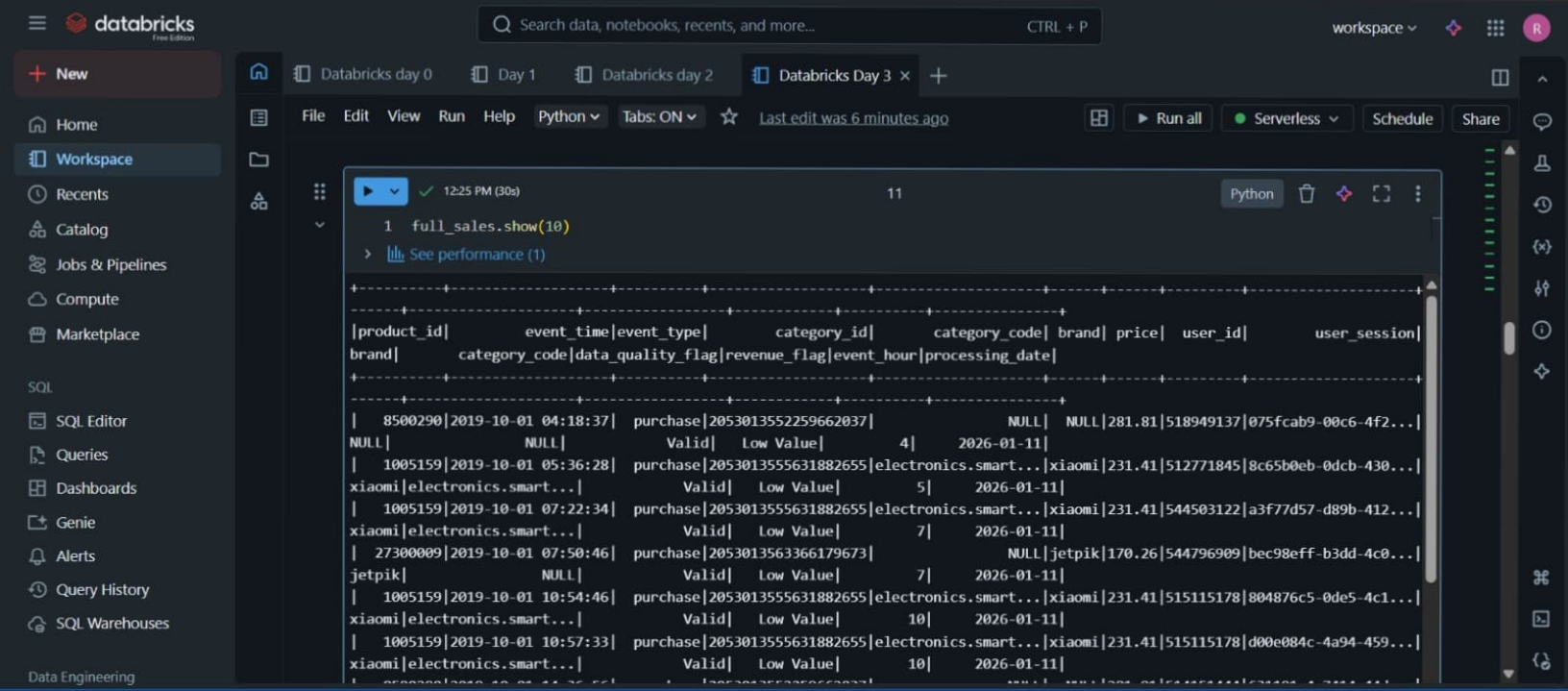
Advanced Full(Outer join)

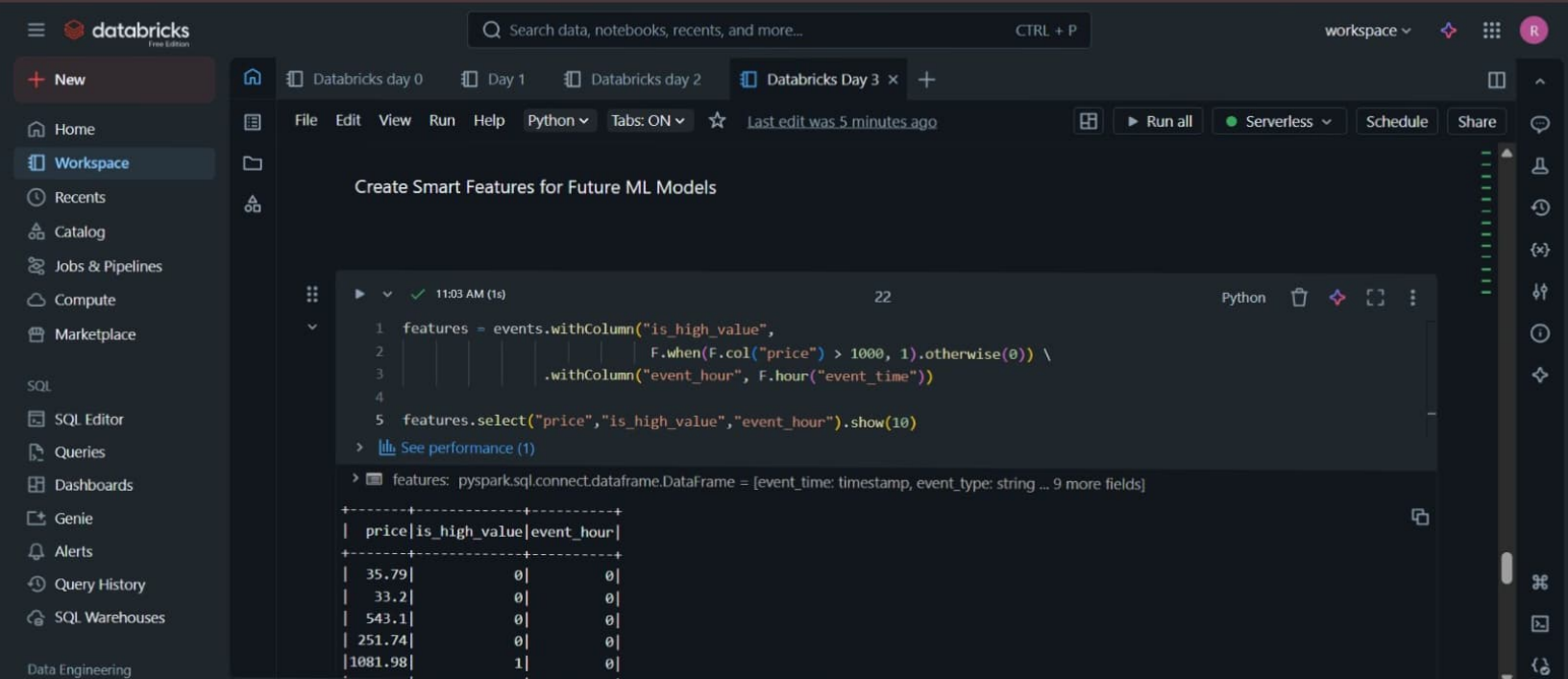
12:24 PM (<1s) 10: Cell 10 Python

```python
import pyspark.sql.functions as F

full_sales = purchases.alias("p") \
    .join(products.alias("pr"), on="product_id", how="outer") \
    .withColumn("data_quality_flag",
                F.when(F.col("p.product_id").isNull(), "Missing in Purchases")
                 .when(F.col("pr.product_id").isNull(), "Missing in Products")
                 .otherwise("Valid")) \
    .withColumn("revenue_flag",
                F.when(F.col("price") > 1000, "High Value")
                 .when(F.col("price") > 500, "Medium Value")
                 .otherwise("Low Value")) \
    .withColumn("event_hour", F.hour("event_time")) \
    .withColumn("processing_date", F.current_date())
```

full_sales: pyspark.sql.connect.dataframe.DataFrame = [product_id: integer, event_time: timestamp ... 13 more fields]


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
File Edit View Run Help Python Tabs: ON
Last edit was 6 minutes ago
Run all
Serverless
Schedule
Share
12:25 PM (30s)
11
Python
1 full_sales.show(10)
See performance (1)
+----------+-------------------+----------+--------------------+-------------------+------+------+---------+--------------------+
------+--------------------+---------------+------------+----------+--------------+
|product_id| event_time|event_type| category_id| category_code| brand| price| user_id| user_session|
brand| category_code|data_quality_flag|revenue_flag|event_hour|processing_date|
+----------+-------------------+----------+--------------------+-------------------+------+------+---------+--------------------+
------+--------------------+---------------+------------+----------+--------------+
| 8500290|2019-10-01 04:18:37| purchase|2053013552259662037| NULL| NULL|281.81|518949137|075fcab9-00c6-4f2...|
NULL| NULL| Valid| Low Value| 4| 2026-01-11|
| 1005159|2019-10-01 05:36:28| purchase|2053013555631882655|electronics.smart...|xiaomi|231.41|512771845|8c65b0eb-0dcb-430...|
xiaomi|electronics.smart...| Valid| Low Value| 5| 2026-01-11|
| 1005159|2019-10-01 07:22:34| purchase|2053013555631882655|electronics.smart...|xiaomi|231.41|544503122|a3f77d57-d89b-412...|
xiaomi|electronics.smart...| Valid| Low Value| 7| 2026-01-11|
| 27300009|2019-10-01 07:50:46| purchase|2053013563366179673| NULL|jetpik|170.26|544796909|bec98eff-b3dd-4c0...|
jetpik| NULL| Valid| Low Value| 7| 2026-01-11|
| 1005159|2019-10-01 10:54:46| purchase|2053013555631882655|electronics.smart...|xiaomi|231.41|515115178|804876c5-0de5-4c1...|
xiaomi|electronics.smart...| Valid| Low Value| 10| 2026-01-11|
| 1005159|2019-10-01 10:57:33| purchase|2053013555631882655|electronics.smart...|xiaomi|231.41|515115178|d00e084c-4a94-459...|
xiaomi|electronics.smart...| Valid| Low Value| 10| 2026-01-11|


databricks
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
File Edit View Run Help Python Tabs: ON
Last edit was 5 minutes ago
Run all
Serverless
Schedule
Share
Create Smart Features for Future ML Models
11:03 AM (1s)
22
Python
features = events.withColumn("is_high_value",
F.when(F.col("price") > 1000, 1).otherwise(0)) \
.withColumn("event_hour", F.hour("event_time"))
features.select("price","is_high_value","event_hour").show(10)
See performance (1)
features: pyspark.sql.connect.dataframe.DataFrame = [event_time: timestamp, event_type: string ... 9 more fields]
+-------+-------------+----------+
| price|is_high_value|event_hour|
+-------+-------------+----------+
35.79	0	0
33.2	0	0
543.1	0	0
251.74	0	0
1081.98	1	0