

DAMG 7370 Designing Advanced Data Architectures for Business Intelligence

Schema Drift

PART 1

Uploading json files to the volume

The screenshot shows a dark-themed user interface for a data catalog or workspace. At the top, there's a search bar with placeholder text "Data, notebooks, recents, and more..." and a keyboard shortcut "CTRL + P". On the right side of the header are "workspace" dropdown, a gear icon, and a profile picture.

The main area is titled "Catalog Explorer > workspace > damg7370 > datastore". Below this, there are tabs: "Overview" (which is selected), "Files", "Details", and "Permissions".

The "Description" section contains a button labeled "AI generate" and a "Add" button.

A breadcrumb navigation bar shows the path: "/Volumes/workspace/damg7370/datastore / schema_drift / demo_smm". To the right of this path is a "Create directory" button.

Below the path is a search bar with the placeholder "Filter files and directories...".

A table lists four JSON files:

Name	Size	Last modified
customer_data_1.json	1.03 KB	50 seconds ago
customer_data_2.json	2.32 KB	1 minute ago
customer_data_3.json	2.09 KB	43 seconds ago
customer_data_4.json	2.17 KB	38 seconds ago

At the bottom left, there's an "About this volume" section with "Owner: prabhakar.roh@northeastern.edu" and a link icon. There's also a "Tags" section with "Add tags" button.

Testing pipeline to see data in bronze

The screenshot shows the Lakeflow Pipelines Editor interface. At the top, there's a header with tabs for 'workspace', 'bronze', 'silver', 'gold', and 'Run pipeline'. Below the header is a 'Pipeline graph' section with a 'Streaming table' node connected to a 'demo_cust_silver_id' node. The 'demo_cust_silver_id' node has an output of 'Output records: 39' and '2 exp'. To the left of the graph is a code editor window titled 'schema drift' containing Python code related to schema drift handling. Below the graph is a table view for the 'demo_cust_silver_sd' table, showing columns: 'PhoneNumber', 'signupDate', '_rescued_data', 'ingestion_datetime', and 'source_filename'. The table contains 11 rows of data.

Rescued_data columns are created in rescued_silver table but no new column is added

This screenshot shows the table view for the 'demo_cust_silver_sd' table. The columns listed are 'id', 'source_filename', '_rescued_data_json_to_map', and '_rescued_data_map_keys'. The table has 42 rows, each showing a different file path from a workspace volume and corresponding null values for the '_rescued_data' columns.

	id	source_filename	_rescued_data_json_to_map	_rescued_data_map_keys
24	/34+UU...	> /Volumes/workspace/damg7370/datas...	null	null
25	734+00...	> /Volumes/workspace/damg7370/datas...	null	null
26	734+00...	> /Volumes/workspace/damg7370/datas...	null	null
27	734+00...	> /Volumes/workspace/damg7370/datas...	null	null
28	734+00...	> /Volumes/workspace/damg7370/datas...	null	null
29	734+00...	> /Volumes/workspace/damg7370/datas...	null	null
30	734+00...	> /Volumes/workspace/damg7370/datas...	null	null
31	734+00...	> /Volumes/workspace/damg7370/datas...	null	null
32	734+00...	> /Volumes/workspace/damg7370/datas...	null	null
33	734+00...	> /Volumes/workspace/damg7370/datas...	null	null
34	734+00...	> /Volumes/workspace/damg7370/datas...	null	null
35	734+00...	> /Volumes/workspace/damg7370/datas...	null	null
36	734+00...	> /Volumes/workspace/damg7370/datas...	null	null
37	974+00...	> /Volumes/workspace/damg7370/datas...	> ["AccountStatus": "Active", "_file_path": "/Volumes/..."]	> ["AccountStatus", "_file_path"]
38	974+00...	> /Volumes/workspace/damg7370/datas...	> ["AccountStatus": "Active", "_file_path": "/Volumes/..."]	> ["AccountStatus", "_file_path"]
39	974+00...	> /Volumes/workspace/damg7370/datas...	> ["AccountStatus": "Suspended", "_file_path": "/Volu..."]	> ["AccountStatus", "_file_path"]
40	974+00...	> /Volumes/workspace/damg7370/datas...	> ["AccountStatus": "Active", "_file_path": "/Volumes/..."]	> ["AccountStatus", "_file_path"]
41	974+00...	> /Volumes/workspace/damg7370/datas...	> ["AccountStatus": "Premium", "_file_path": "/Volum..."]	> ["AccountStatus", "_file_path"]
42	974+00...	> /Volumes/workspace/damg7370/datas...	> ["AccountStatus": "Premium", "_file_path": "/Volum..."]	> ["AccountStatus", "_file_path"]

For add new_columns the column is added directly

me	source_filename	_rescued_data_json_to_map	_rescued_data_map_keys	AccountStatus
4	4.131+00:...	/Volumes/workspace/damg7370/datastore/sche...	null	null
5	4.131+00:...	/Volumes/workspace/damg7370/datastore/sche...	null	null
6	4.131+00:...	/Volumes/workspace/damg7370/datastore/sche...	null	null
7	4.131+00:...	/Volumes/workspace/damg7370/datastore/sche...	null	Active
8	4.131+00:...	/Volumes/workspace/damg7370/datastore/sche...	null	Active
9	4.131+00:...	/Volumes/workspace/damg7370/datastore/sche...	null	Suspended
10	4.131+00:...	/Volumes/workspace/damg7370/datastore/sche...	null	Active
11	4.131+00:...	/Volumes/workspace/damg7370/datastore/sche...	null	Premium
12				null

KEY OBSERVATIONS:

1. **RESCUE MODE (demo_cust_bronze_sd)**:

- AccountStatus appears in _rescued_data as: {"AccountStatus": "Active"}
- Need to query _rescued_data to see the values
- Columns NOT automatically added to table schema

2. **ADDNEWCOLUMNS MODE (demo_cust_bronze_addnew)**:

- AccountStatus appears as actual column with value "Active"
- Can directly SELECT these columns
- _rescued_data is ALWAYS NULL
- Table schema grows automatically

3. WHEN TO USE EACH:

RESCUE MODE :

- Production environments requiring governance
- Need to validate before accepting schema changes
- Want audit trail of schema evolution
- Compliance/regulatory requirements

ADDNEWCOLUMNS MODE :

- Development environments
- Rapid prototyping

- Trusted data sources
- When flexibility > control

PART 2

Catalog Explorer > workspace > damg7370 >

datastore Share Upload to this volume

Overview Files Details Permissions

Description

AI generate Add

/Volumes/workspace/damg7370/datastore / schema_drift / demo_smm

Create directory

Filter files and directories...

Name	Upload summary: 1 file uploaded
customer_data_1.json	
customer_data_2.json	
customer_data_3.json	
customer_data_4.json	(selected)

About this volume

Owner prabhakar.roh@northeastern.edu

Problem in current pipeline:

Streaming DataFrames represent continuous, unbounded data flows with no defined endpoint, while `.collect()` is designed for finite datasets where Spark can read all rows and return them to the driver. When you call `.collect()` on a batch DataFrame, Spark knows there are exactly N rows to materialize and can complete the operation. However, streaming DataFrames are conceptually infinite - data keeps arriving indefinitely, so there's no "last row" to signal completion.

As a result, Spark either throws an error, returns an empty result as a safeguard, or hangs indefinitely waiting for stream termination that never comes. In your original function, the check if not `df.isStreaming` else `[]` evaluates to True for streaming DataFrames, so it returns an empty list `[]`, the for loop iterates zero times, and no columns get added - causing silent failure.

The bronze captures the change in rescued data

	SignupDate	_rescued_data	ingestion_datetime	source_filename
17	2025-01-12	null	2025-11-18T01:19:27.262+00:00	> /Volumes/workspace/damg7370/datastore/sche...
18	2024-07-23	null	2025-11-18T01:19:27.262+00:00	> /Volumes/workspace/damg7370/datastore/sche...
19	4	null	2025-11-18T01:19:27.262+00:00	> /Volumes/workspace/damg7370/datastore/sche...
20	2024-09-18	null	2025-11-18T01:19:27.262+00:00	> /Volumes/workspace/damg7370/datastore/sche...
21	null	{"CreditScore":822,"file_path":"/Volumes/workspace/damg7370/datastore/schemas/CreditScore/822.json"}	2025-11-18T01:23:36.359+00:00	> /Volumes/workspace/damg7370/datastore/sche...
22	null	{"CreditScore":711,"file_path":"/Volumes/workspace/damg7370/datastore/schemas/CreditScore/711.json"}	2025-11-18T01:23:36.359+00:00	> /Volumes/workspace/damg7370/datastore/sche...
23	null	{"CreditScore":610,"file_path":"/Volumes/workspace/damg7370/datastore/schemas/CreditScore/610.json"}	2025-11-18T01:23:36.359+00:00	> /Volumes/workspace/damg7370/datastore/sche...
24	null	{"CreditScore":589,"file_path":"/Volumes/workspace/damg7370/datastore/schemas/CreditScore/589.json"}	2025-11-18T01:23:36.359+00:00	> /Volumes/workspace/damg7370/datastore/sche...
25	null	{"CreditScore":552,"file_path":"/Volumes/workspace/damg7370/datastore/schemas/CreditScore/552.json"}	2025-11-18T01:23:36.359+00:00	> /Volumes/workspace/damg7370/datastore/sche...
26	null	{"CreditScore":510,"file_path":"/Volumes/workspace/damg7370/datastore/schemas/CreditScore/510.json"}	2025-11-18T01:23:36.359+00:00	> /Volumes/workspace/damg7370/datastore/sche...
27	null	{"CreditScore":712,"file_path":"/Volumes/workspace/damg7370/datastore/schemas/CreditScore/712.json"}	2025-11-18T01:23:36.359+00:00	> /Volumes/workspace/damg7370/datastore/sche...
28	null	{"CreditScore":801,"file_path":"/Volumes/workspace/damg7370/datastore/schemas/CreditScore/801.json"}	2025-11-18T01:23:36.359+00:00	> /Volumes/workspace/damg7370/datastore/sche...
29	null	{"CreditScore":520,"file_path":"/Volumes/workspace/damg7370/datastore/schemas/CreditScore/520.json"}	2025-11-18T01:23:36.359+00:00	> /Volumes/workspace/damg7370/datastore/sche...
30	null	{"CreditScore":482,"file_path":"/Volumes/workspace/damg7370/datastore/schemas/CreditScore/482.json"}	2025-11-18T01:23:36.359+00:00	> /Volumes/workspace/damg7370/datastore/sche...

Giving expected columns for the new column to be updated is the solution or to make both the tables batch to do proper schema inference

```
def process_rescue_data_new_fields(df):
    # Hardcode ALL fields that might appear in _rescued_data
    expected_rescued_fields = [
        "CreditScore"
    ]

    # Parse _rescued_data to map
    df = df.withColumn(
        "_rescued_map",
        from_json(
            col("_rescued_data"),
            MapType(StringType(), StringType())
        )
    )

    # Extract each field from _rescued_data
    for field_name in expected_rescued_fields:
        # Only add if column doesn't already exist
        if field_name not in df.columns:
            df = df.withColumn(
                field_name,
                when(
                    col("_rescued_map").isNotNull(),
                    col("_rescued_map")[field_name]
                ).otherwise(None)
            )
    return df
```

Credit Score being populated in the Silver table with corrected code

The screenshot shows a data visualization interface with a table titled "demo_cust_silver_sd". The table has the following columns and data:

	id	signupDate	_rescued_data	ingestion_datetime	source_filename	CreditScore
16	1	2025-01-12	null	2025-11-18T01:19:27.262+00:00	/Volumes/workspace/damg7370/datastore/sche...	null
17	1	2024-07-23	null	2025-11-18T01:19:27.262+00:00	/Volumes/workspace/damg7370/datastore/sche...	null
18	lx344	2024-09-28	null	2025-11-18T01:19:27.262+00:00	/Volumes/workspace/damg7370/datastore/sche...	null
19		2024-09-18	null	2025-11-18T01:19:27.262+00:00	/Volumes/workspace/damg7370/datastore/sche...	null
20			null	2025-11-18T01:23:36.359+00:00	/Volumes/workspace/damg7370/datastore/sche...	822
21			null	2025-11-18T01:23:36.359+00:00	/Volumes/workspace/damg7370/datastore/sche...	711
22			null	2025-11-18T01:23:36.359+00:00	/Volumes/workspace/damg7370/datastore/sche...	610
23			null	2025-11-18T01:23:36.359+00:00	/Volumes/workspace/damg7370/datastore/sche...	589
24			null	2025-11-18T01:23:36.359+00:00	/Volumes/workspace/damg7370/datastore/sche...	552
25			null	2025-11-18T01:23:36.359+00:00	/Volumes/workspace/damg7370/datastore/sche...	510
26			null	2025-11-18T01:23:36.359+00:00	/Volumes/workspace/damg7370/datastore/sche...	712
27			null	2025-11-18T01:23:36.359+00:00	/Volumes/workspace/damg7370/datastore/sche...	801
28			null	2025-11-18T01:23:36.359+00:00	/Volumes/workspace/damg7370/datastore/sche...	520
29			null	2025-11-18T01:23:36.359+00:00	/Volumes/workspace/damg7370/datastore/sche...	482