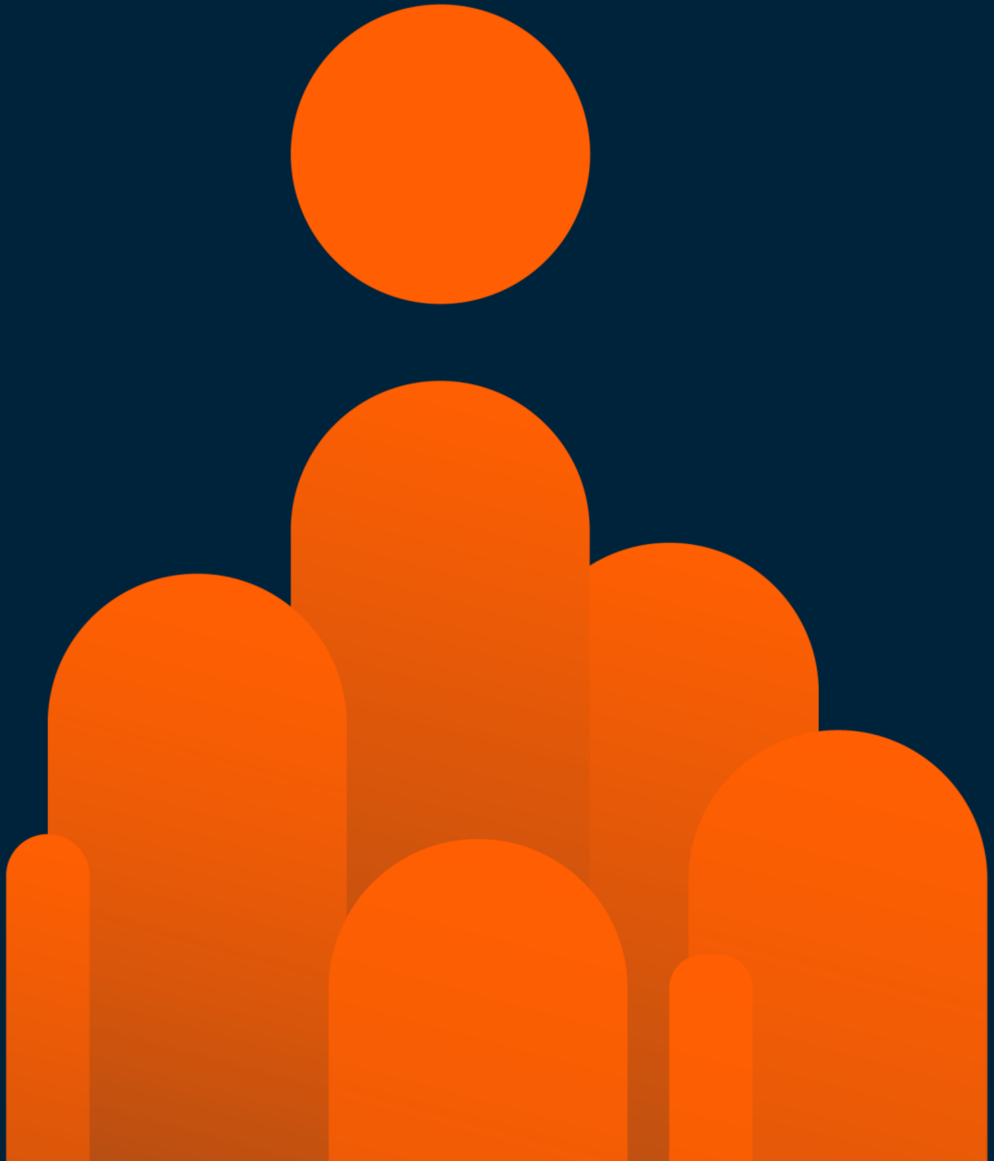
27th Feb 2025

Indian Railways Analysis Use case

Yash Shah
Data Engineering Specialist



USECASE DESCRIPTION

Overview:

The Indian Railways generates vast amounts of data daily, including train details, train schedules, delays, customer satisfaction, and operational performance. Analyzing this data can provide valuable insights to improve efficiency, enhance passenger experience, and optimize resource utilization.

Objective:

To analyze Indian Railways data to identify trends, improve decision-making, and optimize railway operations by leveraging big data processing frameworks like Azure Data Factory, Databricks, and Delta Lake.

SOURCE DATASETS DETAILS

railway_details.csv : **Serves as the base dataset for mapping train operations.**	**delay_details.json :** **Helps in delay pattern analysis and performance improvement.**	**satisfaction_details.json :** **Provides insights into passenger experience and service quality.**
Schema: **Train_id** (String) **Train_name** (String) **Train_color** (String) **Distance** (String) **Src_Station_name** (String) **Dest_Station_name** (String) **Frequency** (String)	**Train_id** (String) **Train_name** (String) **Arrival_time** (String) **Departure_time** (String) **Delay** (String)	**Train_id** (String) **Train_name** (String) **Seats_available** (String) **Cleanliness** (String) **Status** (String) **Satisfaction** (String)

TOOLS USED

Azure Data Lake Storage Gen2

Azure Data Factory

databricks

Azure Logic Apps

ACTIVITIES USED

Lookup

Get Metadata

Copy

Set Variable

Fail

Web e-mail Notification

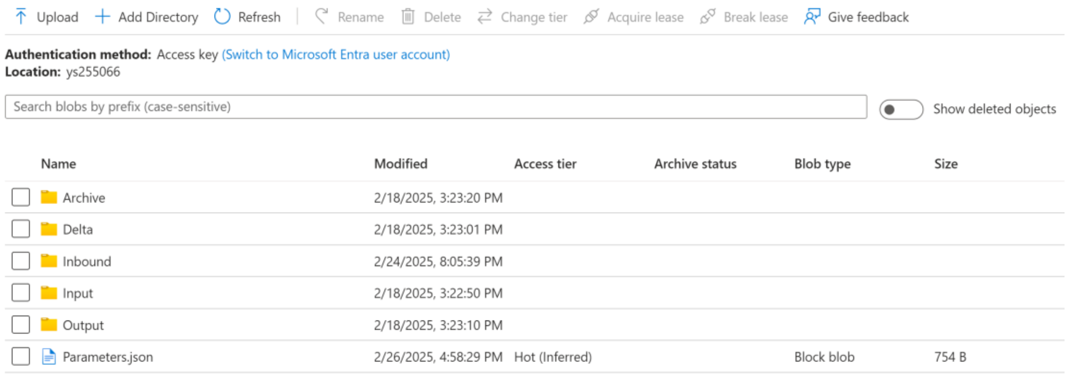
Databricks Notebook

Linked service

Dataset

USECASE DEMO
&
IMPLEMENTATION

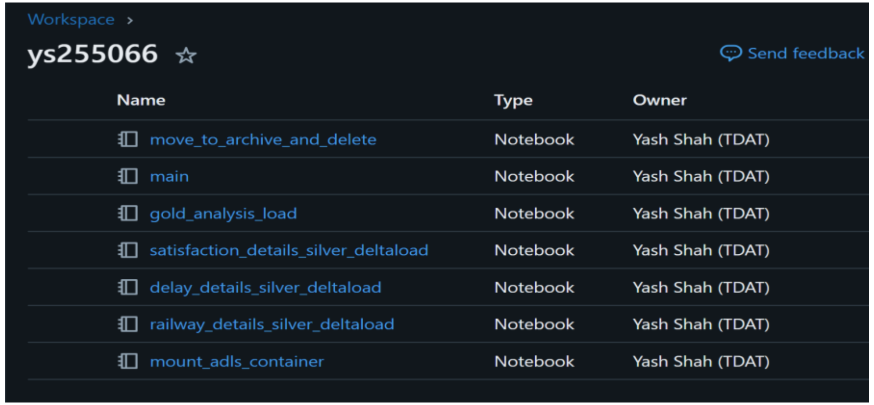
Azure Data Lake Storage Paths



Authentication method: Access key (Switch to Microsoft Entra user account)
Location: ys255066

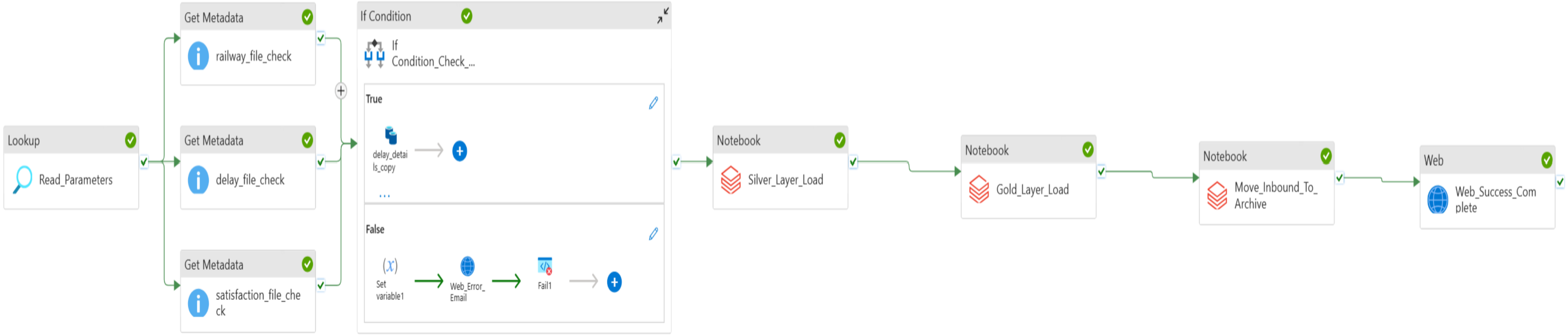
Name	Modified	Access tier	Archive status	Blob type	Size
□ Archive	2/18/2025, 3:23:20 PM				
□ Delta	2/18/2025, 3:23:01 PM				
□ Inbound	2/24/2025, 8:05:39 PM				
□ Input	2/18/2025, 3:22:50 PM				
□ Output	2/18/2025, 3:23:10 PM				
□ Parameters.json	2/26/2025, 4:58:29 PM	Hot (Inferred)		Block blob	754 B

Databricks Workspace



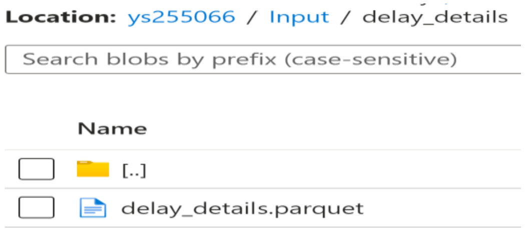
Name	Type	Owner
move_to_archive_and_delete	Notebook	Yash Shah (TDAT)
main	Notebook	Yash Shah (TDAT)
gold_analysis_load	Notebook	Yash Shah (TDAT)
satisfaction_details_silver_deltaload	Notebook	Yash Shah (TDAT)
delay_details_silver_deltaload	Notebook	Yash Shah (TDAT)
railway_details_silver_deltaload	Notebook	Yash Shah (TDAT)
mount_adls_container	Notebook	Yash Shah (TDAT)

Azure Data Factory Pipeline



Staging Layer :-

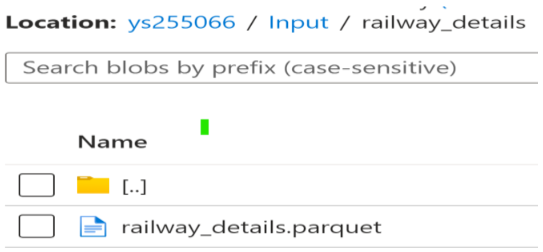
(Source data in Parquet format)



Location: ys255066 / Input / delay_details

Search blobs by prefix (case-sensitive)

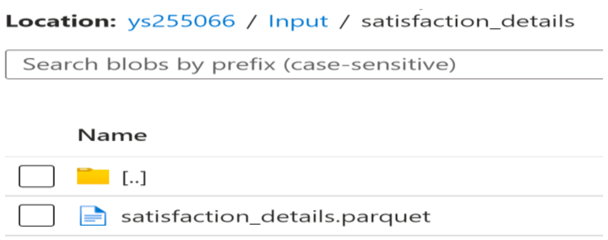
	Name
□	[..]
□	delay_details.parquet



Location: ys255066 / Input / railway_details

Search blobs by prefix (case-sensitive)

	Name
□	[..]
□	railway_details.parquet



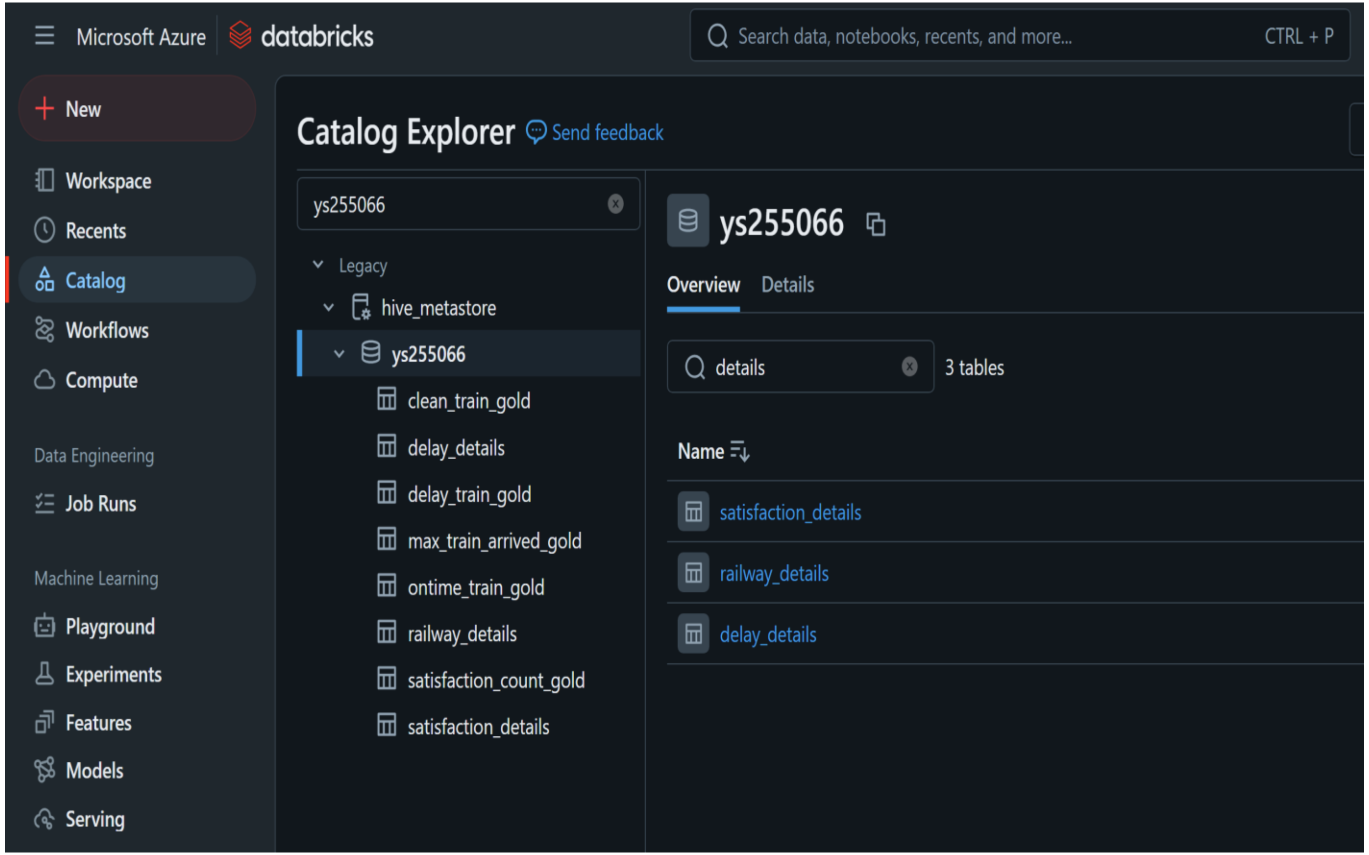
Location: ys255066 / Input / satisfaction_details

Search blobs by prefix (case-sensitive)

	Name
□	[..]
□	satisfaction_details.parquet

Silver Layer :-

(Clean & Transformed data in delta tables format)



ys255066

- Legacy
  - hive_metastore
    - ys255066
      - clean_train_gold
      - delay_details
      - delay_train_gold
      - max_train_arrived_gold
      - ontime_train_gold
      - railway_details
      - satisfaction_count_gold
      - satisfaction_details

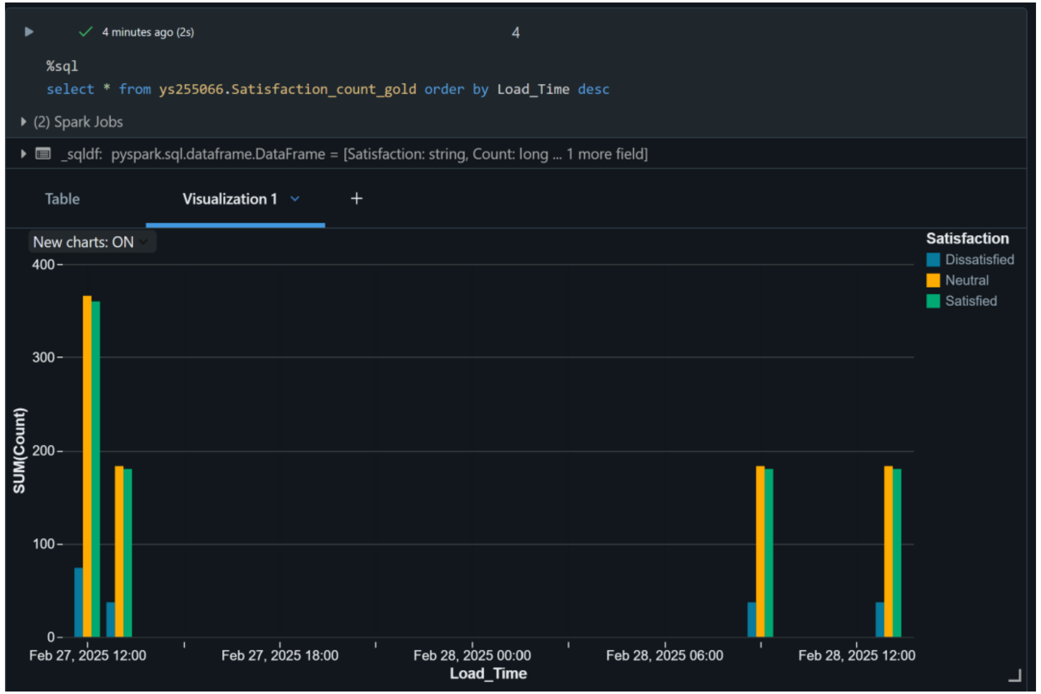
ys255066

details — 3 tables

Name
satisfaction_details
railway_details
delay_details

Gold Layer :- (Stores Aggregated & Business-ready Data)

1. Calculate satisfaction count on basis of satisfied and unsatisfied people



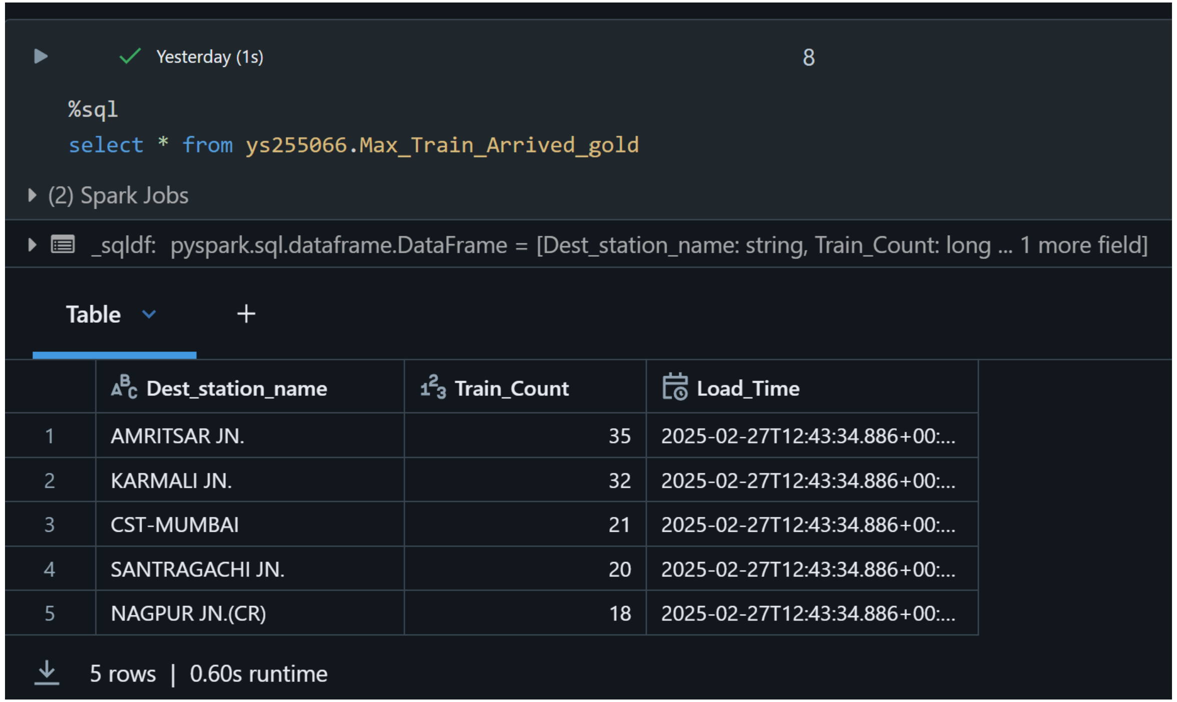
4 minutes ago (2s) 4

```
%sql
select * from ys255066.Satisfaction_count_gold order by Load_Time desc
```

(2) Spark Jobs

_sqldf: pyspark.sql.dataframe.DataFrame = [Satisfaction: string, Count: long ... 1 more field]

2. Find top 5 destinations with maximum train arrivals



Yesterday (1s) 8

```
%sql
select * from ys255066.Max_Train_Arrived_gold
```

(2) Spark Jobs

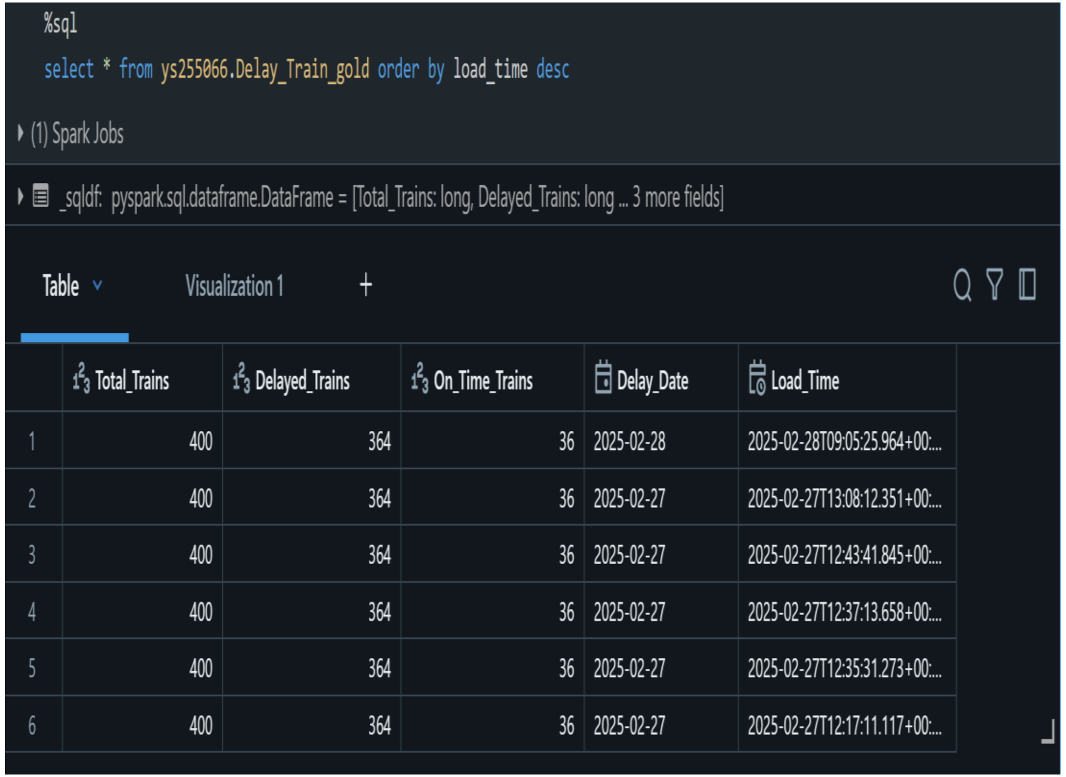
_sqldf: pyspark.sql.dataframe.DataFrame = [Dest_station_name: string, Train_Count: long ... 1 more field]

	Dest_station_name	Train_Count	Load_Time
1	AMRITSAR JN.	35	2025-02-27T12:43:34.886+00:...
2	KARMALI JN.	32	2025-02-27T12:43:34.886+00:...
3	CST-MUMBAI	21	2025-02-27T12:43:34.886+00:...
4	SANTRAGACHI JN.	20	2025-02-27T12:43:34.886+00:...
5	NAGPUR JN.(CR)	18	2025-02-27T12:43:34.886+00:...

5 rows | 0.60s runtime

Gold Layer :- (Stores Aggregated & Business-ready Data)

3. Analyze no. of trains delayed date wise



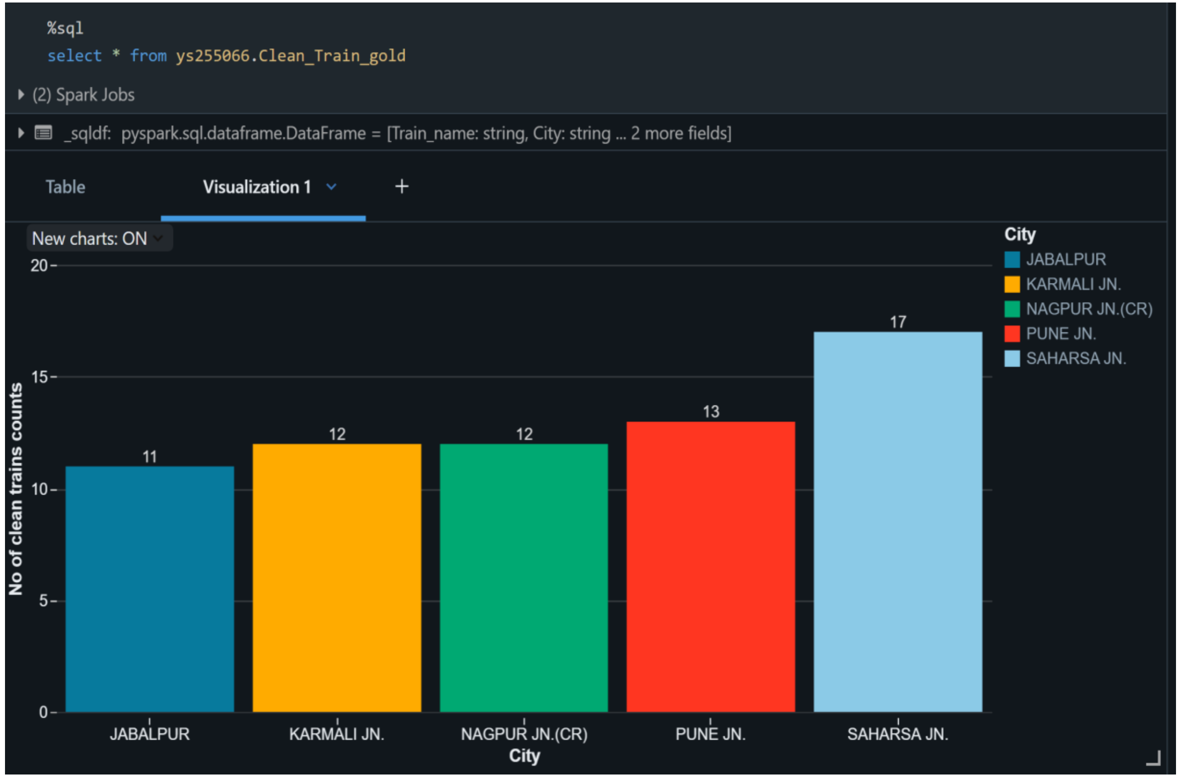
```
%sql
select * from ys255066.Delay_Train_gold order by load_time desc
```

▸ (1) Spark Jobs

▸ _sqldf: pyspark.sql.dataframe.DataFrame = [Total_Trains: long, Delayed_Trains: long ... 3 more fields]

	Total_Trains	Delayed_Trains	On_Time_Trains	Delay_Date	Load_Time
1	400	364	36	2025-02-28	2025-02-28T09:05:25.964+00:...
2	400	364	36	2025-02-27	2025-02-27T13:08:12.351+00:...
3	400	364	36	2025-02-27	2025-02-27T12:43:41.845+00:...
4	400	364	36	2025-02-27	2025-02-27T12:37:13.658+00:...
5	400	364	36	2025-02-27	2025-02-27T12:35:31.273+00:...
6	400	364	36	2025-02-27	2025-02-27T12:17:11.117+00:...

4. Find top 5 cities with Clean trains



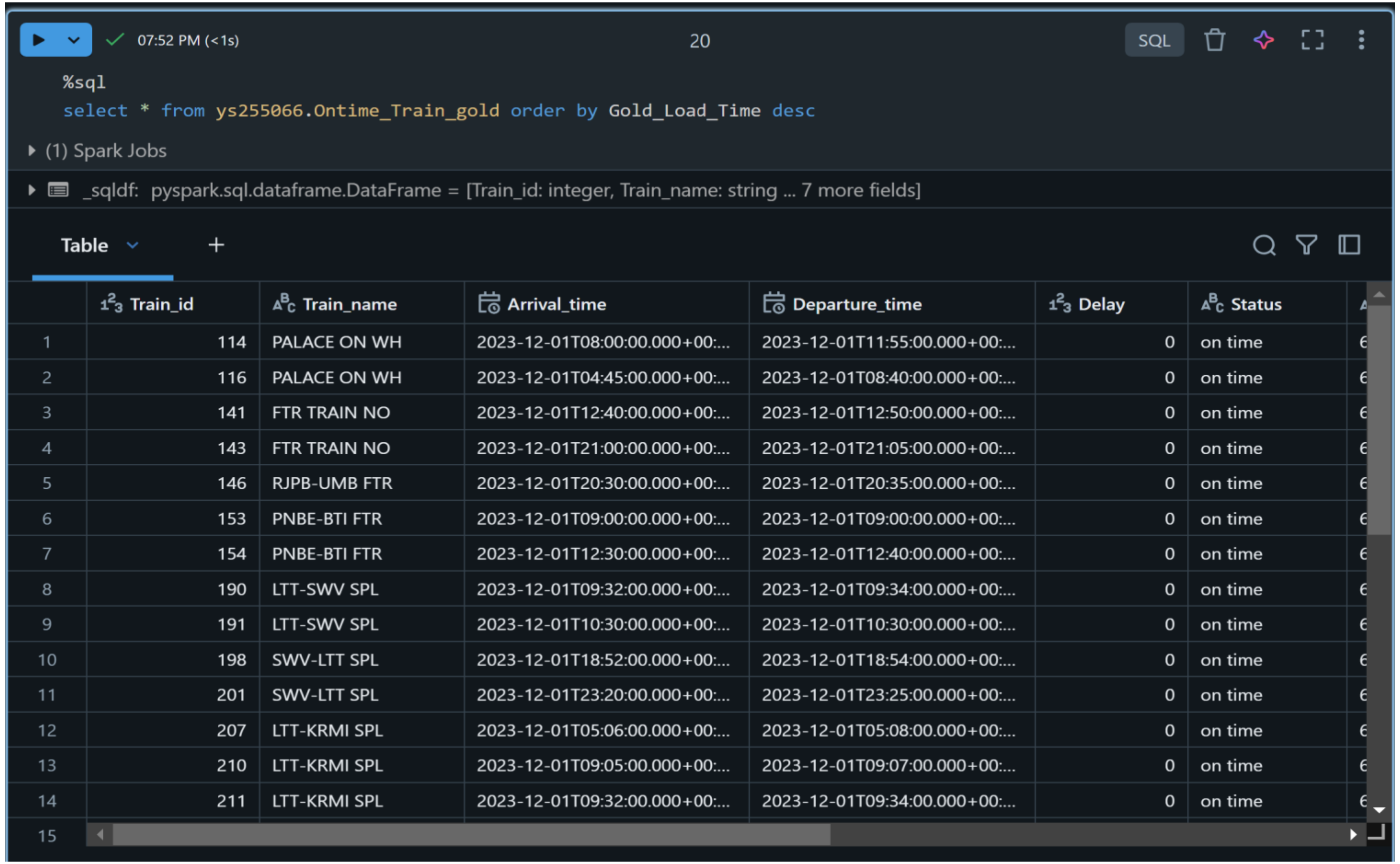
```
%sql
select * from ys255066.Clean_Train_gold
```

▸ (2) Spark Jobs

▸ _sqldf: pyspark.sql.dataframe.DataFrame = [Train_name: string, City: string ... 2 more fields]

Gold Layer :- (Stores Aggregated & Business-ready Data)

5. Identify trains that arrived on time



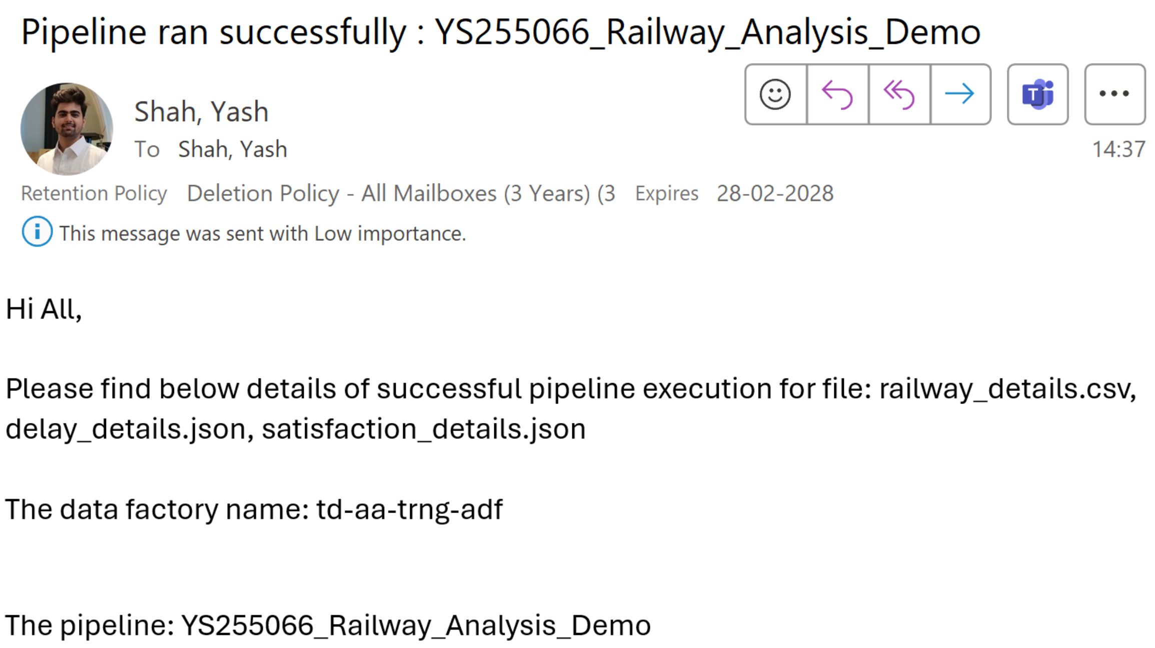
```
%sql
select * from ys255066.Ontime_Train_gold order by Gold_Load_Time desc
```

▸ (1) Spark Jobs

▸ _sqldf: pyspark.sql.dataframe.DataFrame = [Train_id: integer, Train_name: string ... 7 more fields]

	Train_id	Train_name	Arrival_time	Departure_time	Delay	Status	A
1	114	PALACE ON WH	2023-12-01T08:00:00.000+00:...	2023-12-01T11:55:00.000+00:...	0	on time	6
2	116	PALACE ON WH	2023-12-01T04:45:00.000+00:...	2023-12-01T08:40:00.000+00:...	0	on time	6
3	141	FTR TRAIN NO	2023-12-01T12:40:00.000+00:...	2023-12-01T12:50:00.000+00:...	0	on time	6
4	143	FTR TRAIN NO	2023-12-01T21:00:00.000+00:...	2023-12-01T21:05:00.000+00:...	0	on time	6
5	146	RJPB-UMB FTR	2023-12-01T20:30:00.000+00:...	2023-12-01T20:35:00.000+00:...	0	on time	6
6	153	PNBE-BTI FTR	2023-12-01T09:00:00.000+00:...	2023-12-01T09:00:00.000+00:...	0	on time	6
7	154	PNBE-BTI FTR	2023-12-01T12:30:00.000+00:...	2023-12-01T12:40:00.000+00:...	0	on time	6
8	190	LTT-SWV SPL	2023-12-01T09:32:00.000+00:...	2023-12-01T09:34:00.000+00:...	0	on time	6
9	191	LTT-SWV SPL	2023-12-01T10:30:00.000+00:...	2023-12-01T10:30:00.000+00:...	0	on time	6
10	198	SWV-LTT SPL	2023-12-01T18:52:00.000+00:...	2023-12-01T18:54:00.000+00:...	0	on time	6
11	201	SWV-LTT SPL	2023-12-01T23:20:00.000+00:...	2023-12-01T23:25:00.000+00:...	0	on time	6
12	207	LTT-KRMI SPL	2023-12-01T05:06:00.000+00:...	2023-12-01T05:08:00.000+00:...	0	on time	6
13	210	LTT-KRMI SPL	2023-12-01T09:05:00.000+00:...	2023-12-01T09:07:00.000+00:...	0	on time	6
14	211	LTT-KRMI SPL	2023-12-01T09:32:00.000+00:...	2023-12-01T09:34:00.000+00:...	0	on time	6
15							

Pipeline Monitoring With Success & Failure Notifications Over Email.



Pipeline ran successfully : YS255066_Railway_Analysis_Demo

Shah, Yash

To Shah, Yash

14:37

Retention Policy Deletion Policy - All Mailboxes (3 Years) (3 Expires 28-02-2028

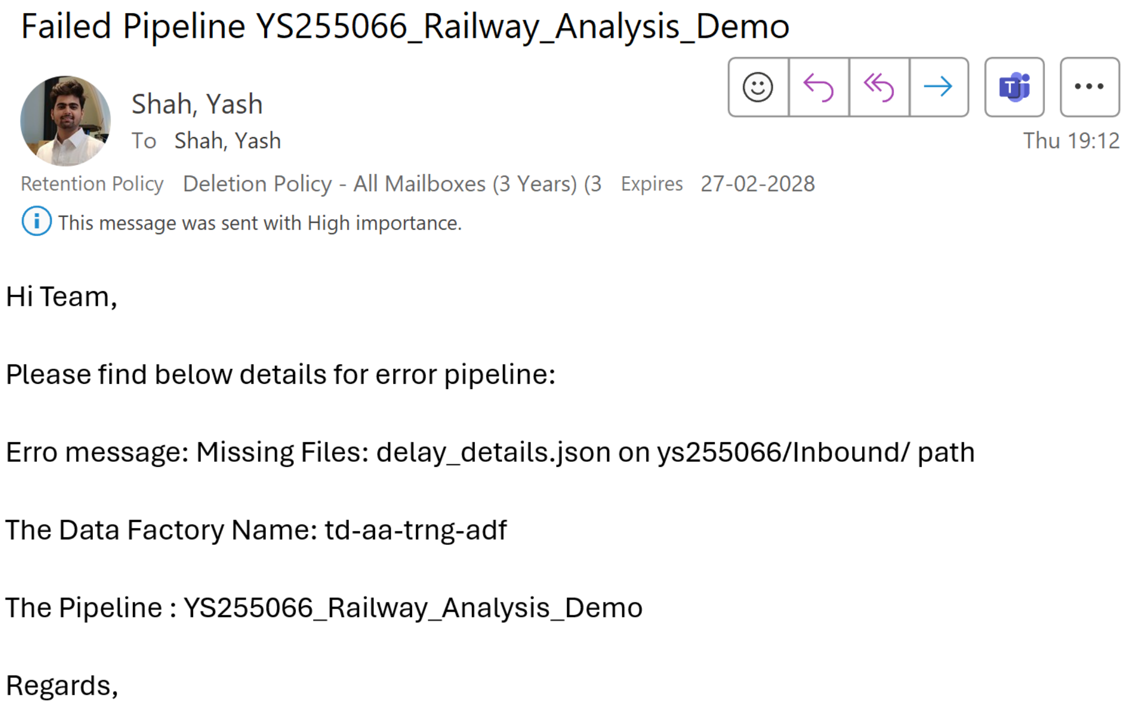
This message was sent with Low importance.

Hi All,

Please find below details of successful pipeline execution for file: railway_details.csv, delay_details.json, satisfaction_details.json

The data factory name: td-aa-trng-adf

The pipeline: YS255066_Railway_Analysis_Demo



Failed Pipeline YS255066_Railway_Analysis_Demo

Shah, Yash

To Shah, Yash

Thu 19:12

Retention Policy Deletion Policy - All Mailboxes (3 Years) (3 Expires 27-02-2028

This message was sent with High importance.

Hi Team,

Please find below details for error pipeline:

Erro message: Missing Files: delay_details.json on ys255066/Inbound/ path

The Data Factory Name: td-aa-trng-adf

The Pipeline : YS255066_Railway_Analysis_Demo

Regards,