ANALYSIS REPORT

EXAM PERFORMANCE ANALYTICS

(Automated using Databricks & AI)

Report By: Sonali Gupta

INDEX

TOOLS USED & TECHNOLOGY STACK

TOOLS USED

The following tools were utilized throughout the development of the Exam Performance Analytics:

- **Databricks –** End-to-end data engineering, analytics, and pipeline orchestration
- **PySpark –** Data transformation and business logic implementation
- **SQL Analytics –** Dashboard queries and KPI generation
- **Databricks Genie AI –** Natural language querying and AI-driven insights

TECHNOLOGY STACK

The project leveraged a modern data analytics and engineering stack including:

- Data Engineering
- ETL (Extract, Transform, Load) Pipelines
- Medallion Architecture (Bronze, Silver, Gold)
- Big Data Processing using PySpark
- Analytics & KPI Development
- AI Integration for Natural Language Insights
- Dashboarding & Visualization

PROJECT OVERVIEW

The Exam Performance Analytics is a modern, automated data analytics solution built using Databricks to process and analyze large-scale examination datasets.

The system enables:

- Automated data ingestion
- Layered data transformation
- Performance evaluation
- KPI generation
- AI-powered insights

The solution follows the Medallion Architecture (Bronze, Silver, Gold) to ensure clean data processing and scalable analytics.

PROBLEM STATEMENT

Educational institutions face challenges in manual exam data processing such as: Slow reporting cycles, High error rates, Limited insight, Lack of intelligent analytics

The goal was to build a comprehensive platform for:

PERFORMANCE ANALYSIS

AUTOMATE PROCESS

AI-DRIVEN ANALYTICS

ORIGINAL DATA DICTIONARY

Student Dataset

Name	Meaning
student_id	Unique student identifier
name	Student name
category	Reservation category
city	Location
marks	Exam score
attendance_status	Present / Absent
attempt_number	Attempt count
exam_date	Exam date

ADDITIONAL TABLES

category_cutoff

feedback_big

DATABRICKS ENVIRONMENT & ARCHITECTURE

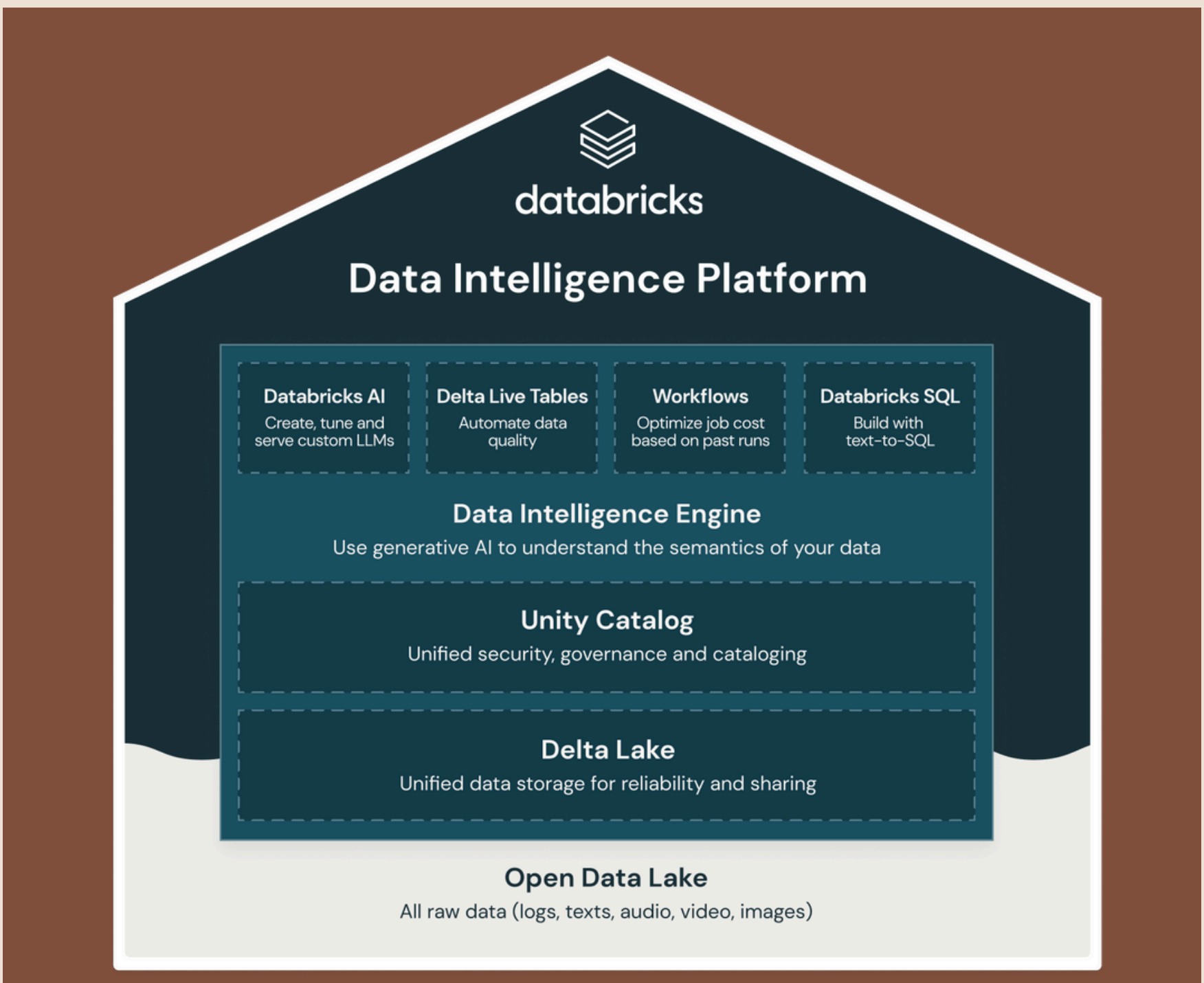
DATABRICKS ENVIRONMENT

All data processing and analytics were implemented using:

- Databricks Platform
- PySpark
- SQL Analytics
- Delta Tables

ARCHITECTURE USED (MEDALLION)

- Bronze Layer – Raw data storage
- Silver Layer – Cleaned and validated data
- Gold Layer – Analytics-ready business tables



PLANNED DATA MODEL

The data was organized into layered analytical tables

Bronze Tables

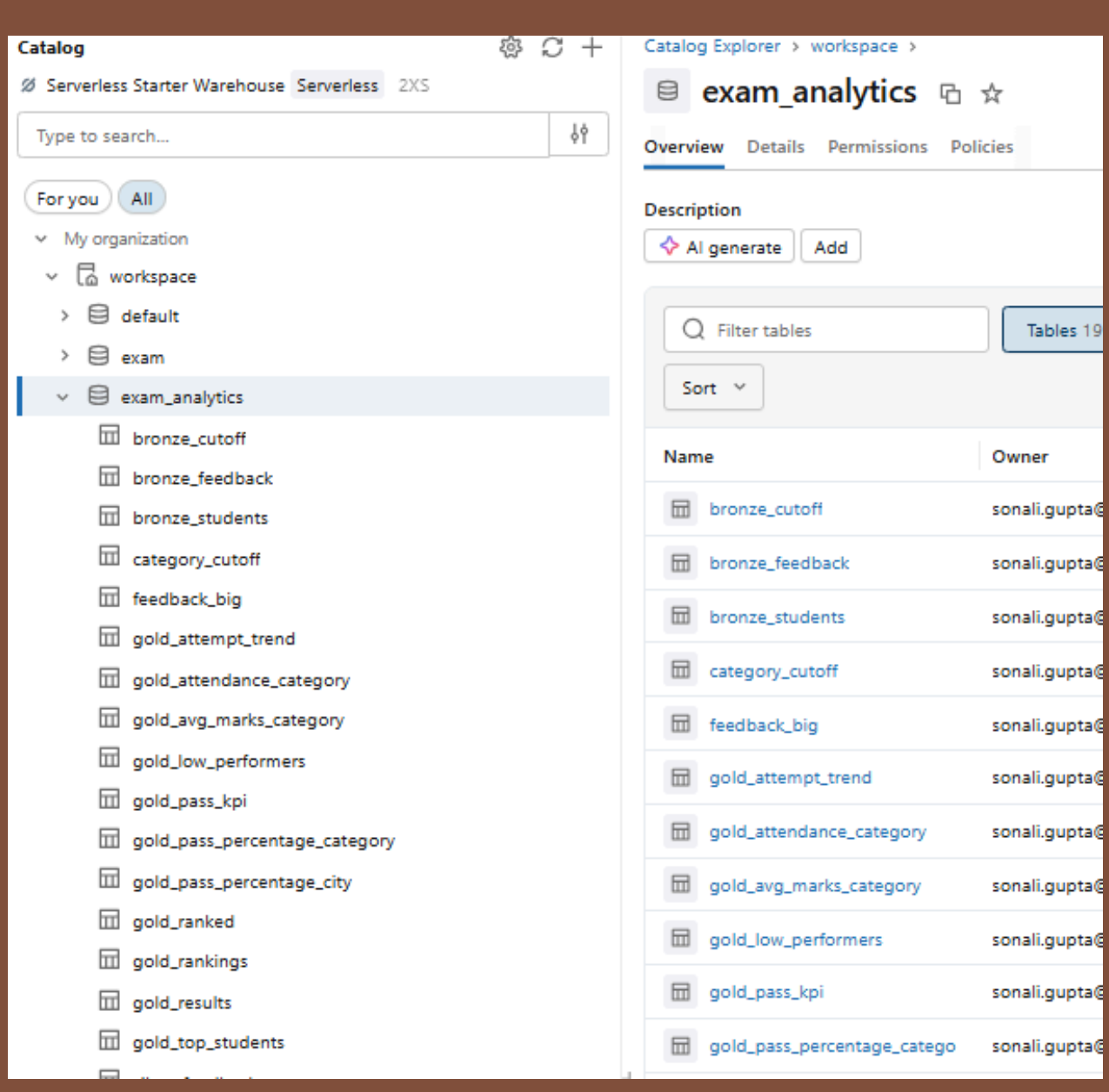
- bronze_students
- bronze_cutoff
- bronze_feedback

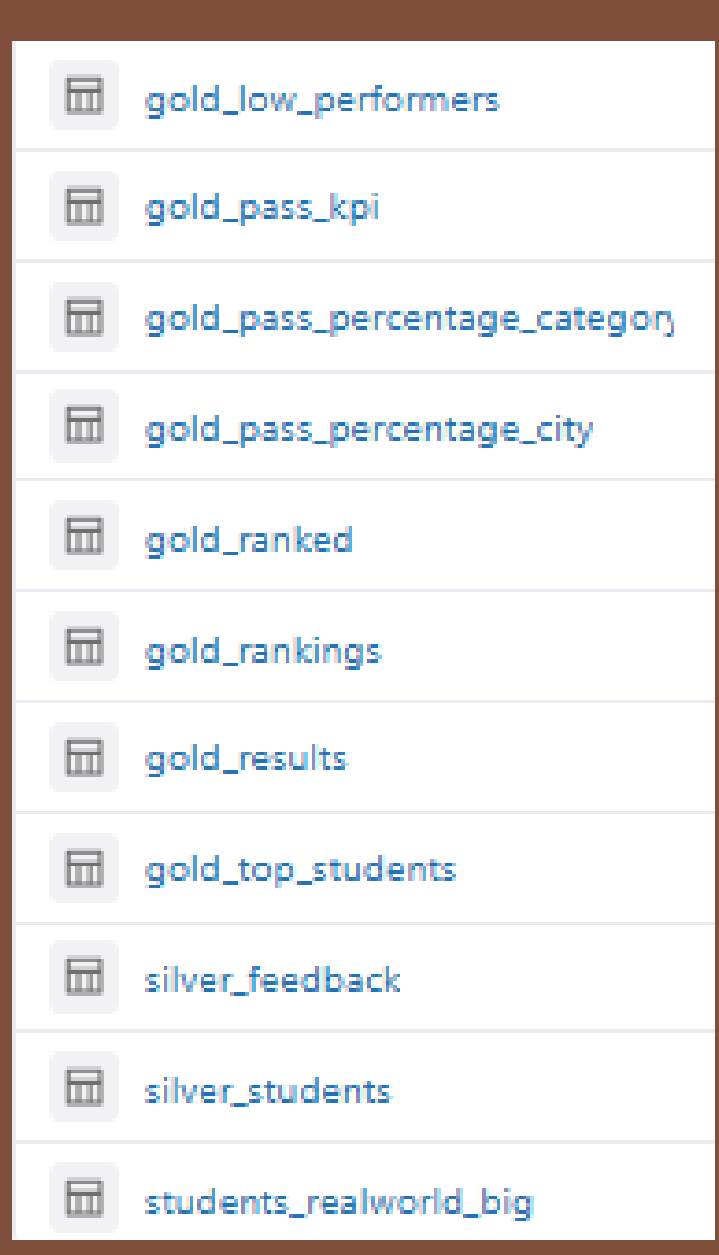
Silver Tables

- silver_students
- silver_feedback

Gold Tables

- gold_results
- gold_rankings
- gold_pass_kpi
- gold_pass_percentage_city
- gold_pass_percentage_category
- gold_avg_marks_category
- gold_attempt_trend
- gold_low_performers
- gold_top_students





DATA INGESTION & CLEANING PROCESS

BRONZE LAYER

Load into Dataframe

2 days ago (20s) 2

```
students_df = spark.table("exam_analytics.students_realworld_big")
cutoff_df   = spark.table("exam_analytics.category_cutoff")
feedback_df = spark.table("exam_analytics.feedback_big")
```

```
students_df: pyspark.sql.connect.dataframe.DataFrame = [student_id: long, name: string ... 6 more fields]
cutoff_df: pyspark.sql.connect.dataframe.DataFrame = [category: string, cutoff_percentage: long]
feedback_df: pyspark.sql.connect.dataframe.DataFrame = [student_id: long, feedback: string]
```

BRONZE LAYER (Raw Storage)

2 days ago (15s) 4

```
students_df.write.mode("overwrite").saveAsTable("exam_analytics.bronze_students")
cutoff_df.write.mode("overwrite").saveAsTable("exam_analytics.bronze_cutoff")
feedback_df.write.mode("overwrite").saveAsTable("exam_analytics.bronze_feedback")
```

Hide performance (3) View all in query history

Statement	Started At	Tasks	Duration	Rows read	Bytes read	Bytes written
L3 feedback_df.write.mode("overwrite")...	Feb 04, 2026, 10:15 AM	0/0 completed	2 s 950 ms	50,000	105.02 KB	104.98 KB
L2 cutoff_df.write.mode("overwrite").s...	Feb 04, 2026, 10:15 AM		3 s 992 ms	4	975 B	932 B
L1 students_df.write.mode("overwrite")...	Feb 04, 2026, 10:15 AM	0/0 completed	7 s 510 ms	50,000	485.50 KB	346.38 KB

- Raw datasets loaded into Databricks tables
- No transformations applied

DATA INGESTION & CLEANING PROCESS

SILVER LAYER

SILVER (Clean Data)

2 days ago (4s) 2

```
## Clean students

from pyspark.sql.functions import col

silver_students = (
    spark.table("exam_analytics.bronze_students")
    .dropna()
    .withColumn("marks", col("marks").cast("int"))
)

silver_students.write.mode("overwrite").saveAsTable("exam_analytics.silver_students")
```

See performance (1)

silver_students: pyspark.sql.connect.dataframe.DataFrame

```
student_id: long
name: string
category: string
city: string
marks: integer
attendance_status: string
attempt_number: long
exam_date: date
```

2 days ago (3s) 3

```
## Clean feedback
silver_feedback = (
    spark.table("exam_analytics.bronze_feedback")
    .dropna()
)

silver_feedback.write.mode("overwrite").saveAsTable("exam_analytics.silver_feedback")
```

See performance (1)

silver_feedback: pyspark.sql.connect.dataframe.DataFrame

```
student_id: long
feedback: string
```

- Null values removed
- Data types standardized
- Consistency checks performed

GOLD LAYER - BUSINESS LOGIC TABLES

GOLD (Business Logic)

✓ 2 days ago (9s) 2

```
from pyspark.sql.functions import col, when

gold_results = (
    spark.table("exam_analytics.silver_students")
    .join(
        spark.table("exam_analytics.category_cutoff"),
        on="category",
        how="left"
    )
    .withColumn(
        "pass_status",
        when(col("marks") >= col("cutoff_percentage"), "Pass")
        .otherwise("Fail")
    )
)

gold_results.write.mode("overwrite").saveAsTable("exam_analytics.gold_results")

display(gold_results.limit(10))
```

See performance (2)

gold_results: pyspark.sql.connect.dataframe.DataFrame = [category: string, student_id: long ... 8 more fields]

Table +

	category	student_id	name	city	marks	attendance_status	attempt_number	exam_date
2	General	2	Neha_2	Mumbai	76	Present	3	2026-01-09
3	General	3	Ritu_3	Delhi	75	Present	1	2026-01-03
4	OBC	4	Neha_4	Bhopal	85	Present	3	2026-02-02
5	General	5	Vikas_5	Lucknow	0	Absent	1	2026-02-15
6	General	6	Nitin_6	Hyderabad	65	Present	1	2026-02-07
7	General	7	Kunal_7	Pune	0	Absent	1	2026-01-28
8	General	8	Vikas_8	Pune	84	Present	1	2026-02-18
9	OBC	9	Suresh_9	Pune	0	Absent	3	2026-01-07
10	General	10	Arjun_10	Delhi	0	Absent	3	2026-02-21

Pass % by Category (KPI)

✓ 2 days ago (1s) 7

```
from pyspark.sql.functions import col, avg

pass_stats = (
    gold_results
    .groupBy("category")
    .agg(
        (avg((col("pass_status") == "Pass").cast("int")) * 100)
        .alias("pass_percentage")
    )
)

display(pass_stats)
```

See performance (1)

pass_stats: pyspark.sql.connect.dataframe.DataFrame = [category: string, pass_percentage: double]

Table +

	category	pass_percentage
1	OBC	45.09375207049626
2	SC	0
3	General	40.875033449290875
4	EWS	48.24100283057015

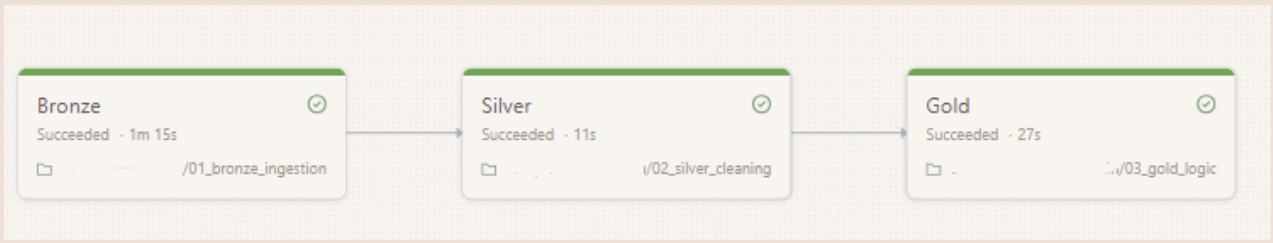
Major transformations included:

- Pass/Fail calculation using category cutoffs
- Merit ranking using dense ranking logic
- KPI aggregations
- Trend analytics

ANALYTICS TABLES SUMMARY

Table Name	Purpose
gold_results	Final student results
gold_rankings	Merit list
gold_pass_percentage_city	City-wise performance
gold_pass_percentage_category	Category performance
gold_avg_marks_category	Academic strength
gold_attempt_trend	Improvement trend
gold_low_performers	At-risk students
gold_top_students	Top 10 performers

DATABRICKS AUTOMATED PIPELINE WORKFLOW



To ensure efficient and repeatable processing, a Databricks Job Workflow was created.

Pipeline Structure:

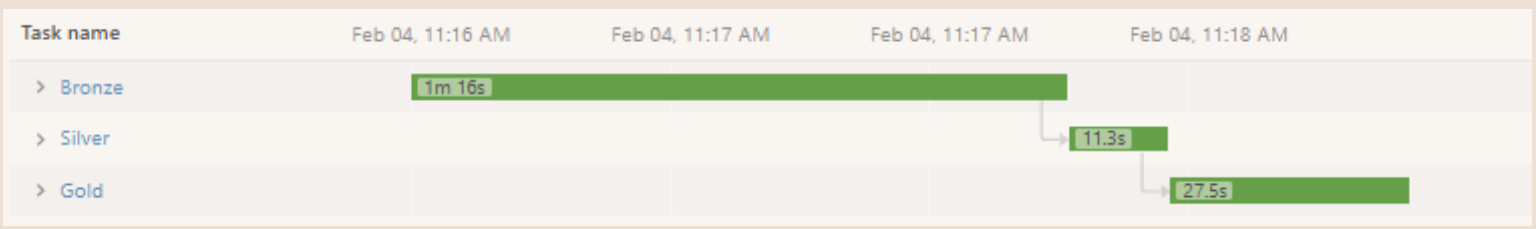
- Bronze Ingestion Notebook
- Silver Cleaning Notebook
- Gold Business Logic Notebook

Each notebook executes sequentially.

Pipeline Execution Highlights:

- Bronze Layer Execution: ~1 minute
- Silver Layer Execution: ~11 seconds
- Gold Layer Execution: ~27 seconds

All tasks completed successfully as shown in the workflow graph and timeline view.



DATABRICKS AUTOMATED PIPELINE WORKFLOW

Benefits of Automation:

- Eliminates manual execution
- Ensures data consistency
- Improves reliability
- Enables scheduled processing

ANALYTICS DASHBOARD DEVELOPMENT

The Exam Performance Dashboard was created using Databricks SQL analytics.

DATA VALIDATION

- Null checks
- Data type verification
- Consistency checks

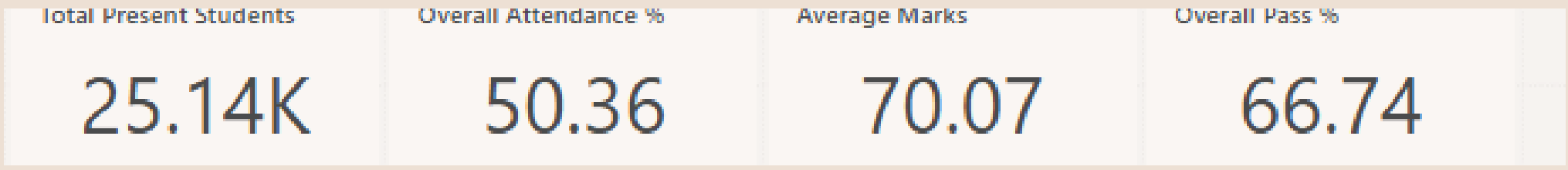
DASHBOARD DEVELOPMENT

All visualizations were built using SQL queries on Gold Layer tables.

DATA EXTRACTION

Gold tables were directly queried in Databricks SQL Warehouse.

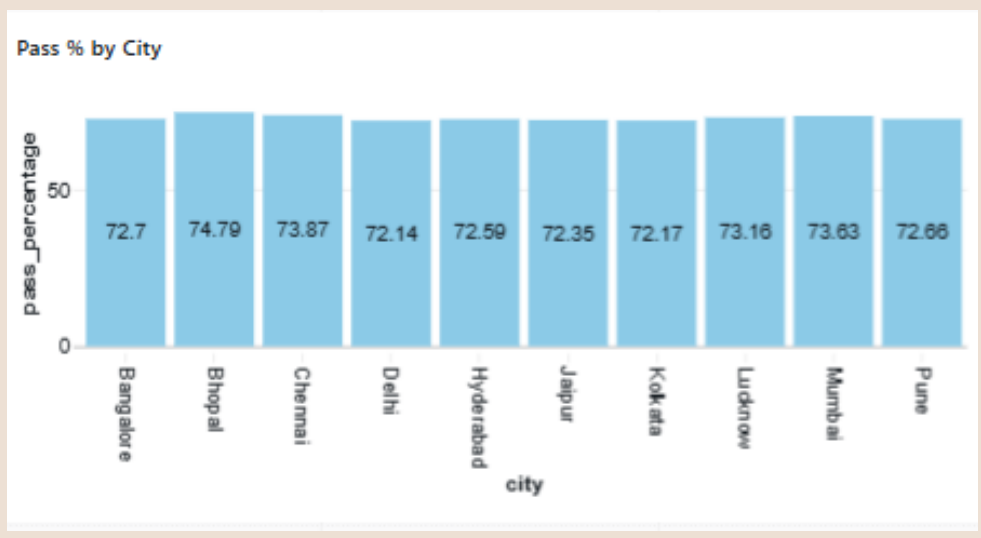
KPIs Displayed:

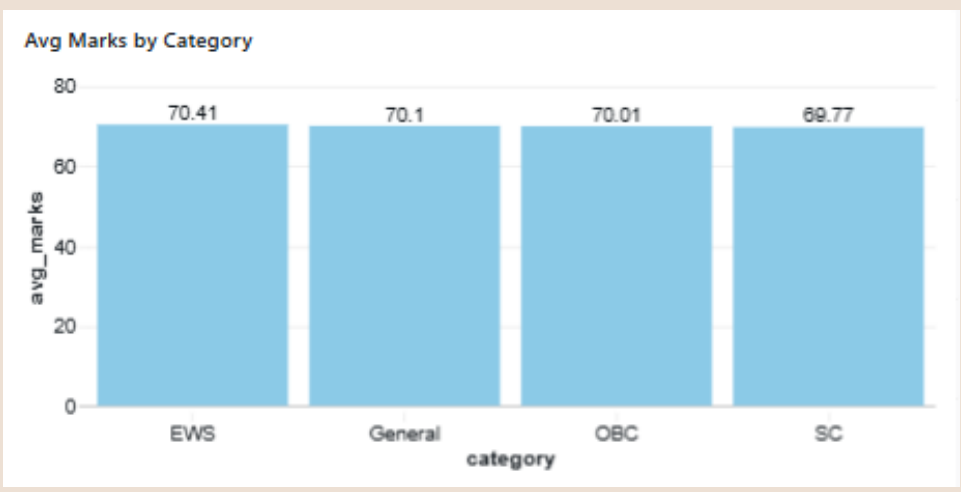


Visual Analytics:

-- Pass % by city--

```
SELECT city, pass_percentage
FROM
exam_analytics.gold_pass_percent
age_city
ORDER BY pass_percentage DESC;
```





-- Avg Marks by Category --

```
SELECT *
FROM
exam_analytics.gold_avg_marks_ca
tegory
ORDER BY avg_marks DESC;
```

-- Top Student --

```
SELECT student_id, name, city, marks, rank
FROM exam_analytics.gold_top_students
ORDER BY rank
limit 10;
```

Top 10 Students

student_id	name	city	marks	rank
985	Meera_985	Lucknow	95	1
3850	Ritu_3850	Pune	95	1
1321	Kiran_1321	Kolkata	94	2
977	Tarun_977	Bangalore	94	2
3936	Suresh_3936	Kolkata	93	3
1011	Komal_1011	Lucknow	93	3
5210	Ritu_5210	Bangalore	93	3
1723	Isha_1723	Chennai	92	4
2092	Pradeep_2092	Lucknow	92	4
3844	Manish_3844	Hyderabad	92	4

AI INTEGRATION USING DATABRICKS GENIE

Databricks Genie was integrated to enable natural language interaction with exam datasets.

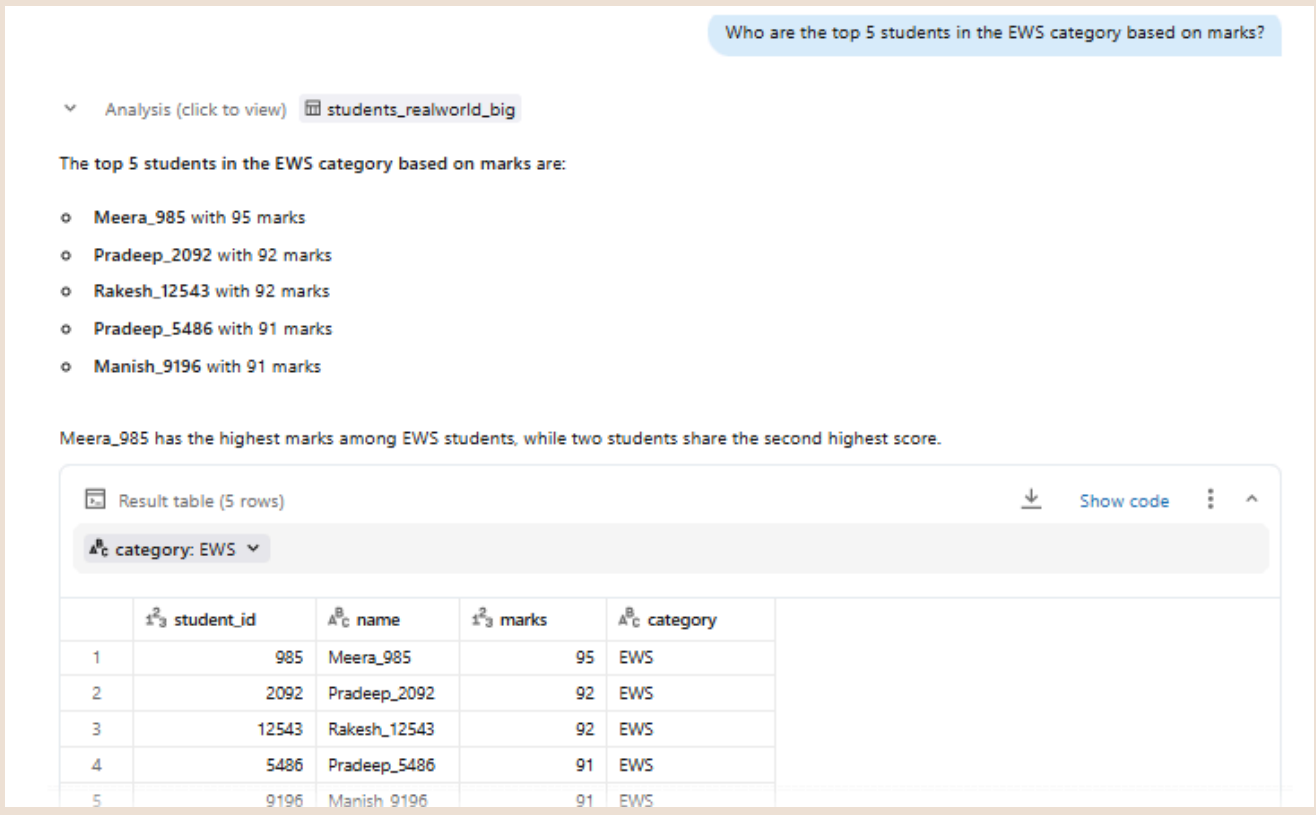
Sample Genie Questions:

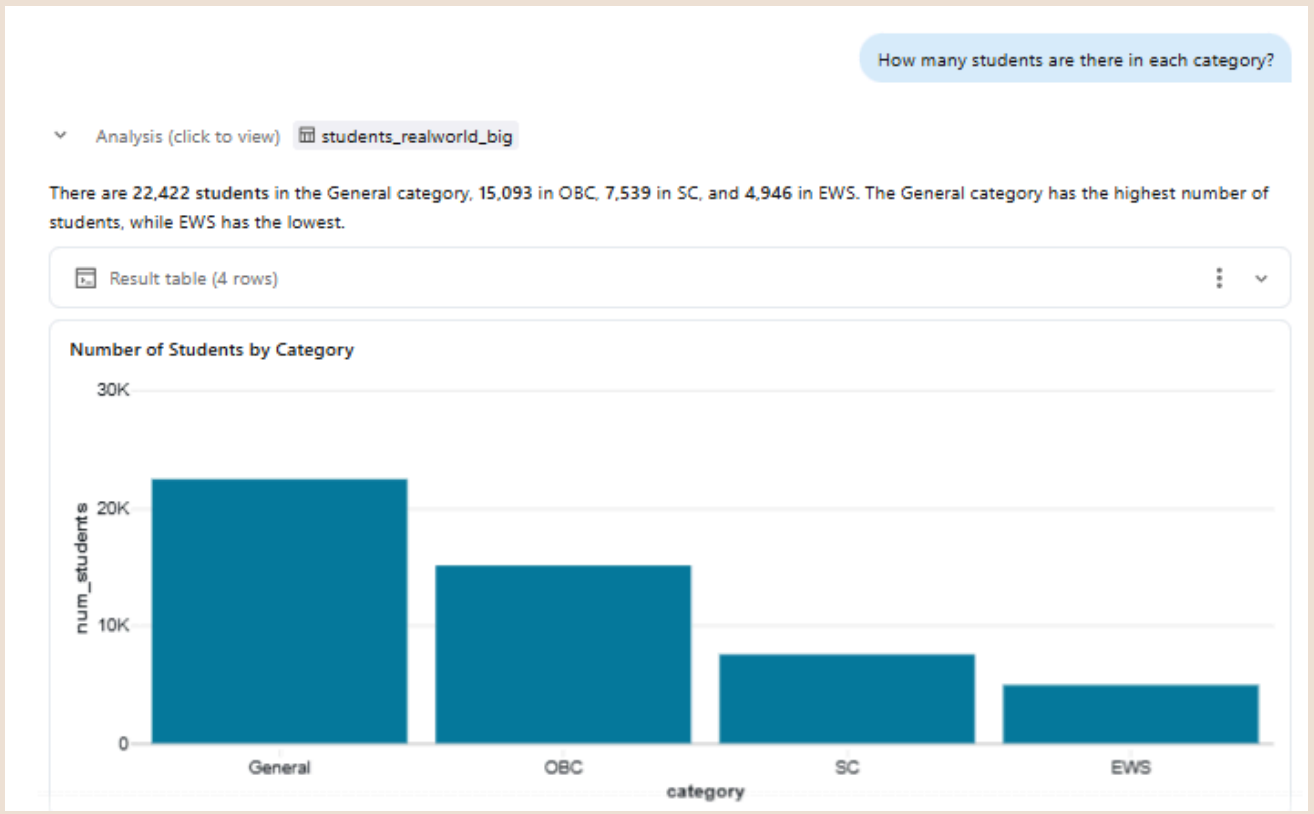
“How many students are there in each city?”

“Who is the top student according to marks?”

“Who is the lowest performer with attendance present?”

Genie automatically converted these questions into optimized SQL queries and displayed both tabular results and visual charts.

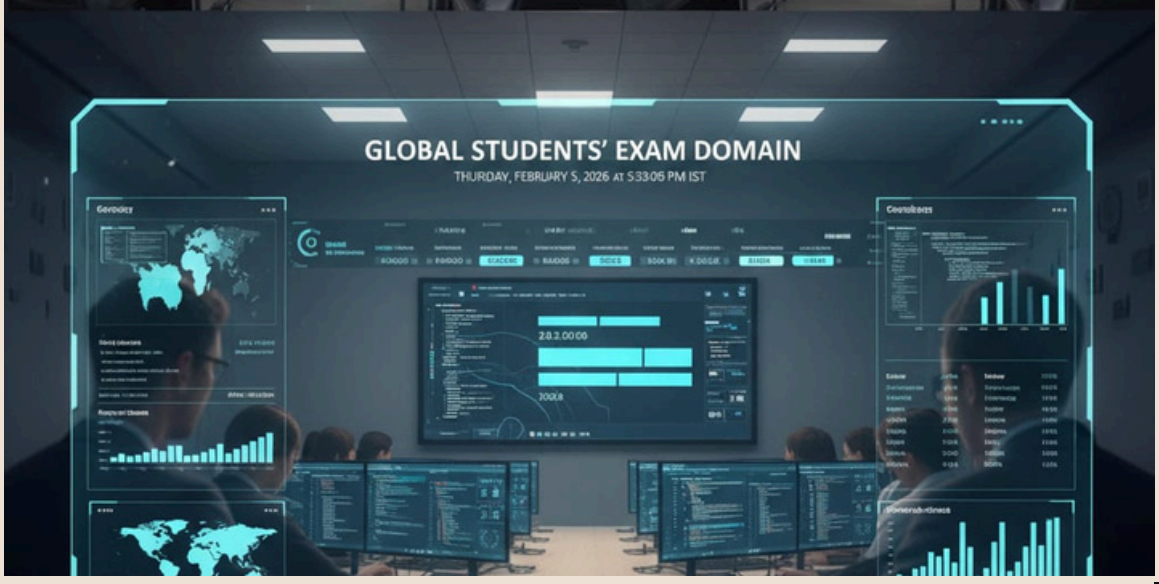




DATABRICKS GENIE AND DATA MODEL

Advantages of Genie AI

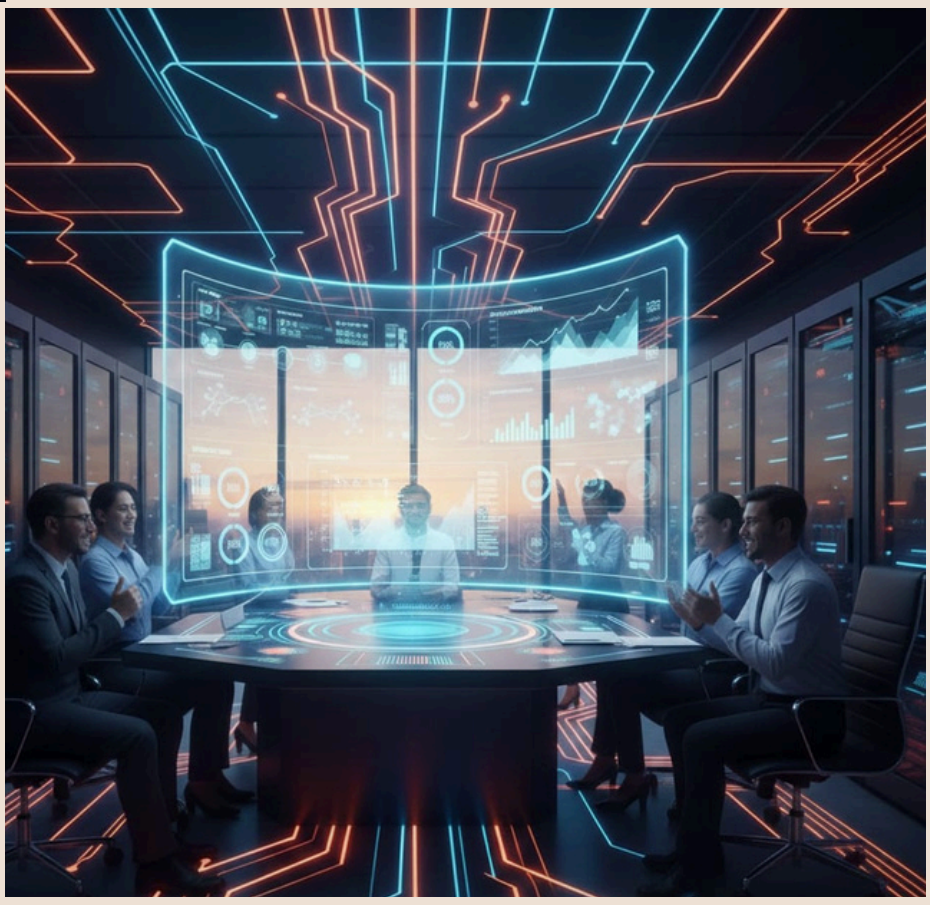
- No SQL knowledge required
- Faster decision-making
- Business-friendly analytics
- AI-generated visualizations



FINAL DATA MODEL

The Medallion Architecture structured all analytical workflows ensuring:

- Clean data lineage
- Optimized performance
- Easy scalability



INSIGHTS & CONCLUSIONS

The Databricks dashboard revealed crucial performance trends:

Performance Trends

- Student scores improve with multiple attempts
- Clear variation across cities

Category Performance

- Certain categories outperform consistently
- Cutoff logic ensures fair evaluation

Risk Identification

- Low-performing students flagged automatically

Merit Insights

- Top students identified instantly

Key Insights

• The highest-performing student across all categories is **Meera_985 (Student ID: 985)**, who achieved the top score of 95 marks, making her the overall topper of the examination.

• The lowest-performing student among those who were present for the exam is **Neha_19674** (Student ID: 19674), who scored only 3 marks, highlighting a case requiring immediate academic intervention.

• A significant performance concern was identified, with **1,331 students** who attended the exam scoring **below 50 marks**, indicating a considerable portion of students are underperforming despite being present.

• Within the OBC category, the top-performing student is **Suresh_3936** (Student ID: 3936), who secured 93 marks, demonstrating strong academic performance in this group.

• In the **EWS category**, the top five performers were led by **Meera_985 (95 marks), followed by Pradeep_2092 and Rakesh_12543** with 92 marks each, and **Pradeep_5486 and Manish_9196** with 91 marks, reflecting competitive performance within this category.

• Category-wise student distribution shows that the **General category has the highest enrollment with 22,422 students**, followed by OBC (15,093), SC (7,539), and EWS (4,946), indicating demographic variations in participation.

• City-wise analysis highlights that **Lucknow has the highest student participation with 5,081 students**, while **Delhi recorded the lowest count with 4,897 students**, with other major cities such as Mumbai, Hyderabad, and Bangalore showing comparable participation levels.

• Performance comparison across categories indicates that the **EWS category** demonstrates the **highest average marks**, making it the best-performing category overall in terms of academic outcomes.

INSIGHTS & CONCLUSIONS

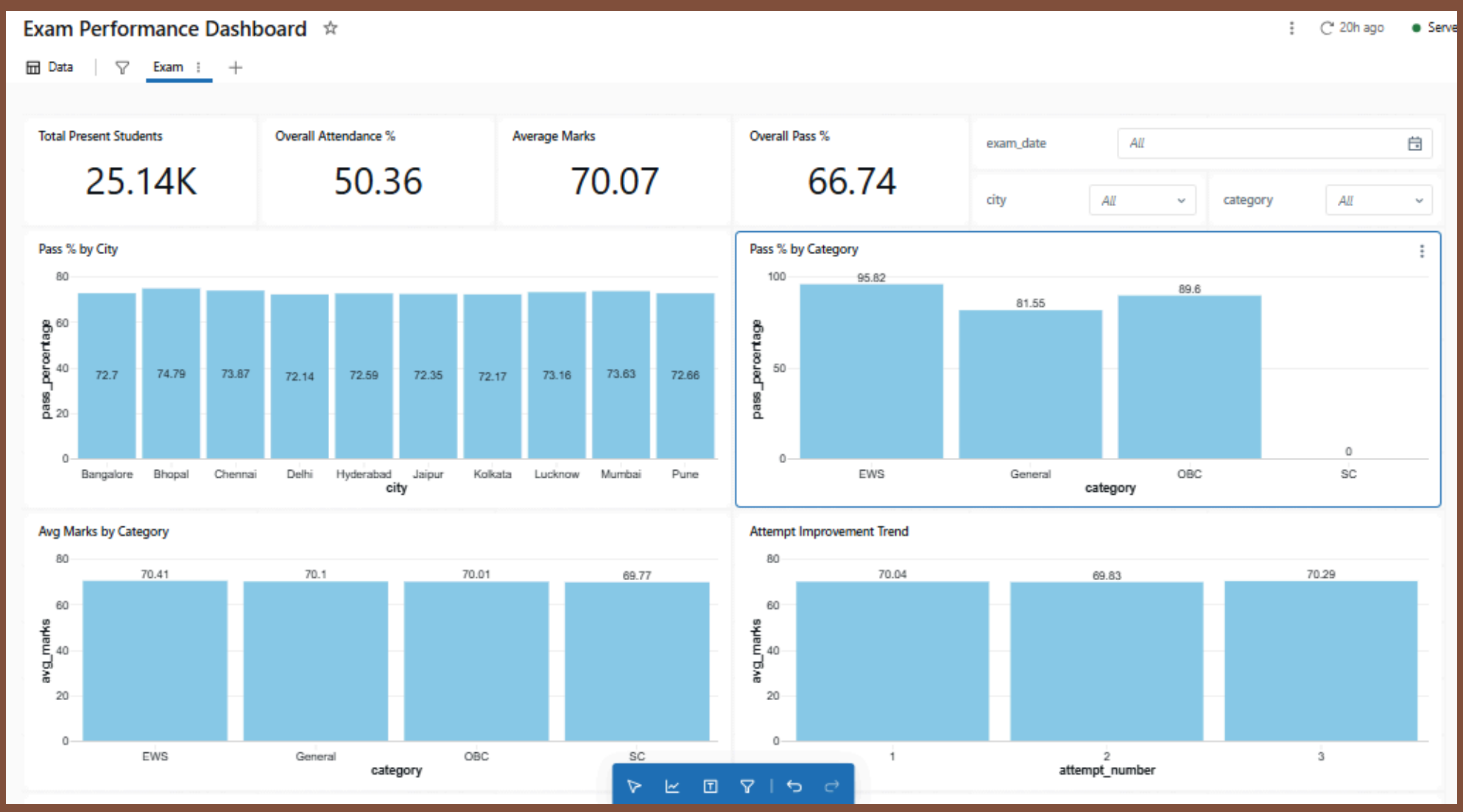
Conclusion

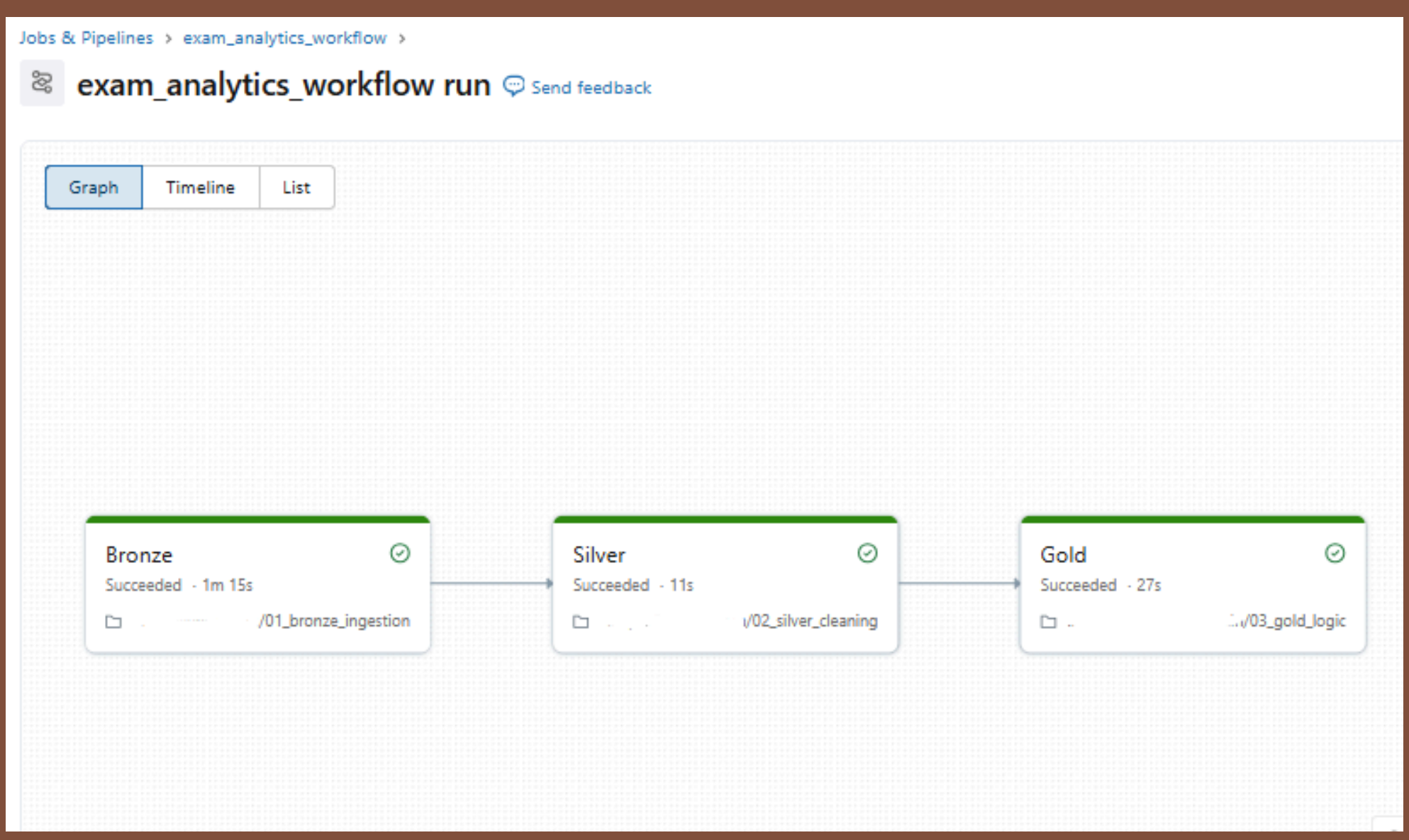
The project successfully implemented:

- End-to-end Databricks ETL pipeline
- Automated Medallion architecture
- KPI-driven analytics
- AI-powered natural language querying

The solution replaced manual Excel-based reporting with a scalable, intelligent analytics platform.

DASHBOARD & PIPELINE SNIPPETS





THANK YOU

Report By: Sonali Gupta