**Incremental and Automated Data Engineering Pipeline for Aviation Analytics Using Databricks Lakehouse**

**PROJECT REPORT**

submitted by

Aniket Tripathi

**to**

The National Institute of Electronics & IT, Chennai in partial fulfilment of the requirements for the program

***PG Program in Data Engineering***



**Data Science Group**

**National Institute of Electronics and Information Technology, Chennai**

(An Autonomous Scientific Society of Ministry of Electronics and Information Technology, Government of India)

No.: 25, Gandhi Mandapam Road, Chennai – 600025, Tamil Nadu

[www.nielit.gov.in/chennai](http://www.nielit.gov.in/chennai)

**September, 2025**

**DECLARATION**

I undersigned hereby declare that the project report “**Incremental and Automated Data Engineering Pipeline for Aviation Analytics Using Databricks Lakehouse**”, submitted for partial fulfilment of the requirements of the program ‘**PG program in Data Engineering**’ at National Institute of Electronic & IT, Chennai is a bonafide work done by our team consisting of Mr. Aniket Tripathi under supervision of **Mr. Sourav Acherjee** . This submission represents our ideas in our own words and where ideas or words of others have been included, We have adequately and accurately cited and referenced the original sources. We also declare that We have adhered to academic honesty and integrity ethics and have not misrepresented or fabricated any data, idea, fact, or source in my submission. We understand that any violation of the above will cause disciplinary action by the institute and/or the University and can also evoke penal action from the sources that have thus not been properly cited or from whom proper permission has not been obtained. This report has not previously formed the basis for awarding any degree, diploma or similar title of any other University.

Place: New Delhi

Date: 08/09/2025 Aniket Tripathi

**CERTIFICATE**

This is to certify that project work entitled **Incremental and Automated Data Engineering Pipeline for Aviation Analytics Using Databricks Lakehouse**  submitted by **Aniket Tripathi** in partial fulfillment of the requirements for the PG program in Data Engineering at NIELIT Chennai is a bonafide record of the work carried out by them under my guidance and supervision from to 10/07/2025 to 28/07/2025.

|  |  |
| --- | --- |
| Mr. Sourav Acherjee  Scientist ‘B’ & Course Coordinator  NIELIT Chennai | Mr. Ishant Kumar Bajpai  Scientist ‘D’ & Group Head  NIELIT Chennai |

**Date: 08/09/2025**

**ACKNOWLEDGEMENT**

The completion of any project depends upon cooperation, coordination and combined efforts of several sources of knowledge. This report acknowledges a number of guidance, supervision, stimulation and lot of inspiration from numerous people. I would like to express my gratitude to all who helped us directly or indirectly in the successful completion of our project work.

I express my gratitude to **Dr. Sanjeev Kumar Jha**, Director I/C, NIELIT, Chennai for his persistent support.

I express my sincere gratitude to **Mr. Ishant Kumar Bajpai**, Scientist ‘D’ & Group Head for his guidance, encouragement and consistent support in completing the project successfully.

I wish to thank to **Mr. Sourav Acherjee**, Scientist ‘B’ & Course Coordinator, and all the staff of the **Data Science** Group for their valuable feedback and support.

# ABSTRACT

This research project illustrates the conceptualization and execution of a comprehensive data engineering pipeline by employing **Databricks** as the integrated data platform and **dbt (data build tool)** for systematic data transformation and modeling. The main aim of this project is to replicate a practical, industry-standard data engineering workflow that ingests, processes, transforms, and delivers data for analytical and reporting purposes in a scalable and automated fashion.

Data is collected from raw sources and subsequently ingested into **Databricks** where it is housed within an integrated Delta Lakehouse architecture. The project employs Databricks for distributed data processing, enforcement of data quality, and for orchestration tasks using **Lakeflow Declrative Pipeline**, and employs **dbt** for composing reusable, versioned SQL transformation models. This layered approach aligns with the latest best practices, and therefore ensures traceability, modularity, and performance optimization.

With the development of durable ETL/ELT pipelines, incremental data loading strategies, and dimensional models clearly defined, the project delivers data sets ready for analytics, thus enabling data-driven decision-making. Also, the integration of Databricks with dbt showcases the cooperative ability of cloud-native solutions for bridging data engineering and analytics, thus reducing complexity and maximizing team productivity.

## CONTENTS

|  |  |
| --- | --- |
| **Contents** | **Page No.** |
| ACKNOWLEDGEMENT | 4 |
| ABSTRACT | 5 |
| LIST OF FIGURES  LIST OF TABLES | 7  8 |
| Chapter 1. Introduction  Chapter 2. Literature Survey | 9  11 |
| Chapter 3. Methodology | 12 |
| Chapter 4. Hardware Tools and Component | 23 |
| Chapter 5. Software Tools and Languages | 25 |
| Chapter 6. Implementation | 26 |
| Chapter 7. Results and Discussions | 29 |
| Chapter 8. Conclusion and Future Scope | 31 |
| REFERENCE | 33 |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.**  1  2  3  4 | **Title**  Bronze Pipeline  Silver Pipeline  Gold Layer Execution  Project Architecture | | **Page No.**  **13-14**  **15-16**  **18-21**  **27** |
|  |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **No.** | **Title** | **Page No.** |
| 1 | Literature Survey | 11 |
| 2 | Hardware Tools and Requirements | 22-23 |
| 3 | SOFTWARE TOOLS USED | 24 |
| 4 | Results | 29-30 |

**CHAPTER 1**

**INTRODUCTION**

The exponential growth of data in the modern digital economy has dramatically transformed the working and decision-making paradigm of organizations. Businesses no longer rely on traditional reporting systems alone instead, organizations are now seeking sophisticated, automated, and scalable data engineering solutions capable of an integrating multiple data sources, upholding data integrity, and delivering analytics-ready datasets for enabling real-time decision-making.

Organizations of today are confronted with data that is vast, heterogeneous, and everchanging. In order for such a problem to be managed effectively, data engineering became the foundation of the analytics paradigm since it offers the necessary pipelines and the architecture needed for the data to be accessible, reliable, and actionable. At the core of such development has been the incorporation of cloud-native platforms and the transformation methods that simplify the complete data lifecycle.

The project, “ **Building Scalable ETL Pipelines with Azure Databricks and dbt**”, shows the entire approach brought together under one roof by integrating the Databricks Lakehouse Platform and the data build tool (dbt). Databricks provides an enterprise-grade cloud environment for distributed data storage and data processing through Delta Lake, and dbt introduces modularity, version control, and transformation-as-code practices into the pipeline. They complement each other and offer an end-to-end workflow that enables ingestion, transformation, modeling, orchestration, and analytics delivery.

To add more dynamism, automation, and practical applications alignment for the pipeline, the project integrates:

1. Dynamic Notebooks allowing unprocessed data ingestion into the Bronze layer, thus ensuring flexible and reusable ingestion workflows across different datasets.
2. Lakeflow Declarative Pipelines(formerly Delta Live Tables (DLT)) for managing Silver layer transformations, declarative pipeline execution, automatic data quality checks, and incremental refreshes.
3. dbt Models to develop the fine-grained Gold layer transformations, imposing modularity and versioning for analytics-ready data sets.

The project also focuses on industry best practices like:

1. Medallion Architecture (Bronze–Silver–Gold layers) for the management of data quality.
2. Incremental data loading serves to enhance performance while simultaneously minimizing processing overhead.
3. Orchestration and workflow automation for minimal human intervention.

The integration of Dynamic Notebooks, DLT, and dbt in the project not only exemplifies the technological prowess ingrained in the Databricks-native tools but also signifies the evolution of contemporary workflows for enterprise data platforms. The project develops an enterprise-compatible, scalable, and production-quality infrastructure for the needs of data-driven organizations today.

**CHAPTER 2**

**LITERATURE SURVEY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Literature** | **Year** | **Technology** | **Insights** |
| 1 | Databricks. The Medallion Architecture in Lakehouse Design. Databricks Blog | 2022 | Databricks Lakehouse, Delta Lake | Introduces the Bronze, Silver, and Gold layered architecture to ensure traceability and usability. |
| 2 | dbt Labs. dbt Documentation and Best Practices | 2023 | dbt (Data Build Tool) | Introduced modular, version-controlled SQL models with built-in testing and documentation. Adopted in the Gold layer for analytics-ready transformations |
| 3 | Databricks. Delta Live Tables (DLT) Documentation | 2023 | Delta Live Tables (DLT) | Provided declarative, automated pipeline execution with data quality expectations and incremental refresh. Used for Silver layer transformations. |
| 4 | Inmon, W. H. Building the Data Warehouse (4th ed.) | 2005 | Enterprise Data Warehousing | Established principles of centralized data management, governance, and integration strategies. Inspired ingestion and quality validation practices. |
| 5 | Kimball, R., & Ross, M. The Data Warehouse Toolkit (3rd ed.) | 2013 | Dimensional Modeling, Star Schema | Explained dimensional modeling for efficient analytical queries. Gold layer dbt models follow this star schema approach. |

**CHAPTER 3**

# METHODOLOGY

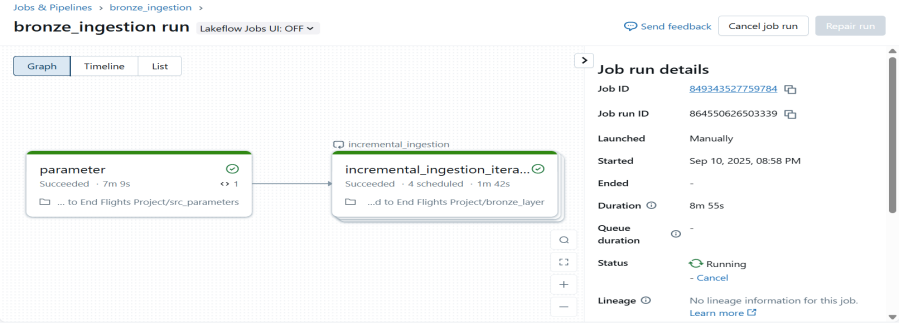
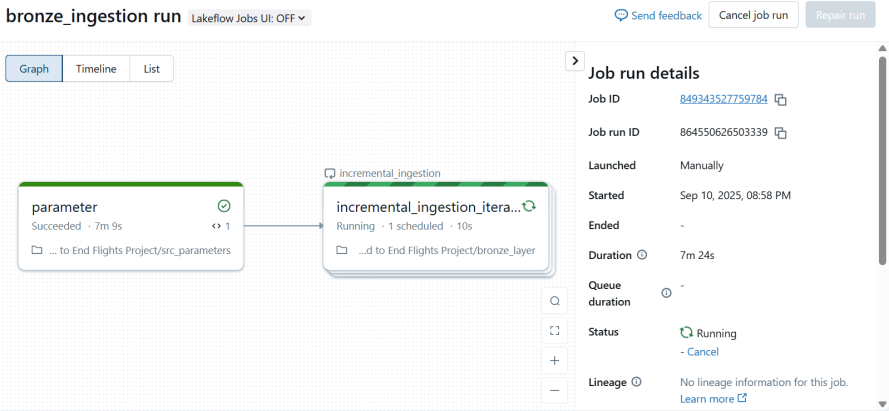
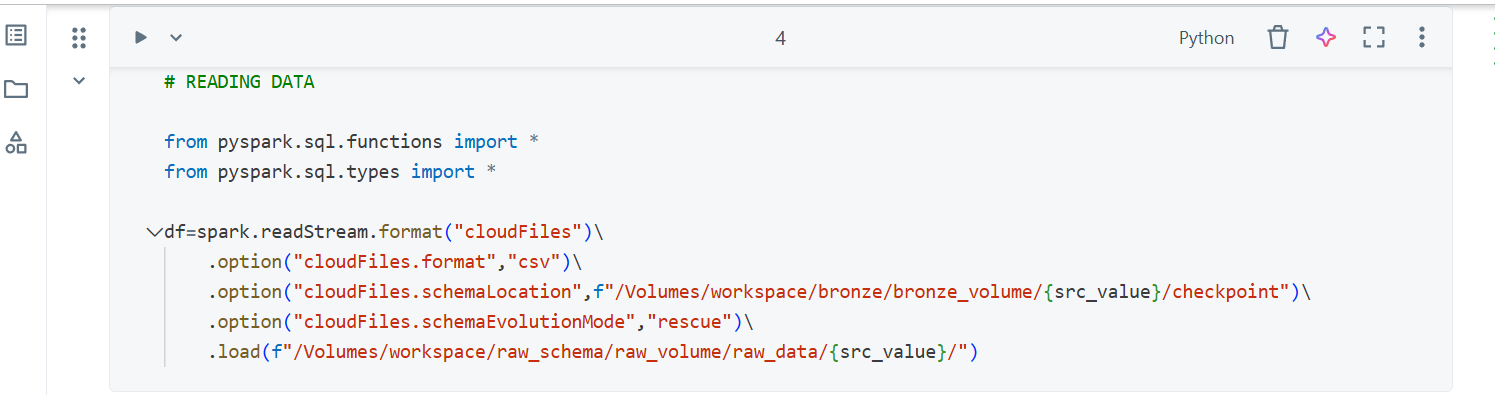
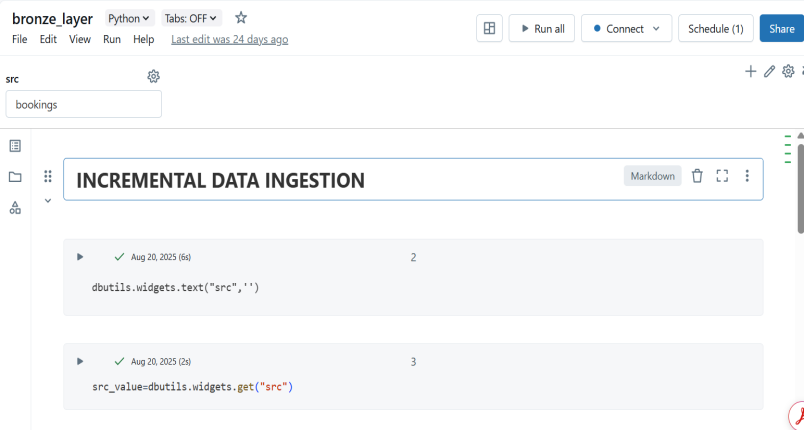
The methodology followed in this project is a systematic approach to designing and implementing a modern data engineering pipeline on Databricks, with dbt for modular transformations. The process ensures scalability, reliability, and analytics-readiness of the data. The following stages were adopted:

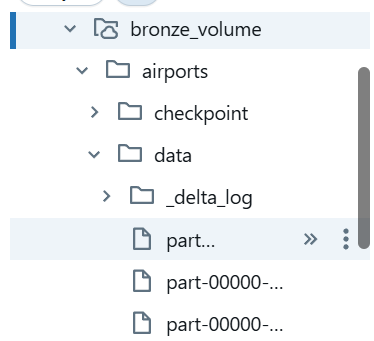
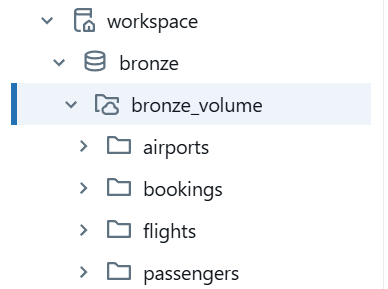
1. **Requirement Analysis**

The project began with a study of data requirements and reporting objectives. Source data sets were reviewed in order to identity key entities, such as passengers, airports, flights and bookings data. The business requirements emphasized needing to draw insights on the basis of trends in Passenger booking,Flight occupancy rates and delays,Seasonal and route-based travel insights thus shaping the data modeling and pipeline development approach.

**2. Data Acquisition/Extraction(Bronze Layer)**

1. Data was ingested in bronze layer using Dynamic notebooks in Databricks
2. To enable incremental and continuous ingestion, **Databricks Autoloader** was used. This allowed the pipeline to automatically detect new files arriving in the data source (e.g., bookings, flights, passengers data) and load them without reprocessing the entire dataset.
3. Autoloader also handles schema information by supporting schema inference and schema evolution. This allows the pipeline to automatically adapt to changes in incoming data structures.
4. Parameterized notebooks ensured flexibility for handling multiple datasets such as airports, flights, passengers, and bookings.
5. Data was stored in its raw state in **Delta format**, ensuring ACID compliance, time-travel, and full lineage tracking





**3. Data Transformation (Silver Layer)**

The Silver Layer was designed to transform raw ingested data into clean, standardized, and analytics-ready datasets. This layer was implemented using Databricks Declarative Lakeflow Pipelines(formerly Delta Live Tables (DLT)) with a Python-based pipeline file.

Key Activities Performed:

1. Data Cleaning and Transformation:

* Dropped the \_rescued\_data column temporarily created during ingestion
* Performed Data Type Casting(ex- converting flight\_date to DateType)
* A modified\_date column was incorporated into each table to document the precise processing timestamp

1. DLT Pipeline Development:

* Created streaming views (e.g., trans\_bookings, trans\_flights, trans\_passengers, trans\_airports) in order to apply above transformations
* Utilized the @dlt.table and @dlt.view decorators to explicitly define tables and views within the pipeline code
* Implemented data quality checks with @dlt.expect\_all, ensuring critical fields (e.g., booking\_id, passenger\_id) are not null

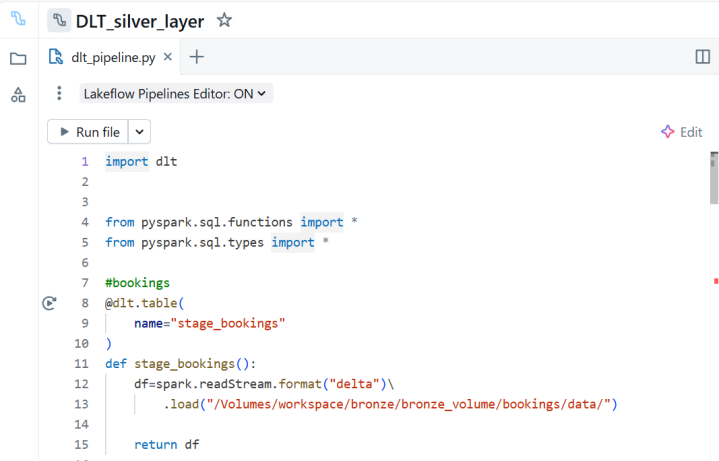
1. Slowly Changing Dimensions (SCD) and Change Data Capture (CDC)

* Made use of dlt.create\_auto\_cdc\_flow for automatic change data capture and managing Slowly Changing Dimensions (Type 1)
* This ensured core objects (airports, passengers, flights) were updated appropriately in the Silver layer without any duplication

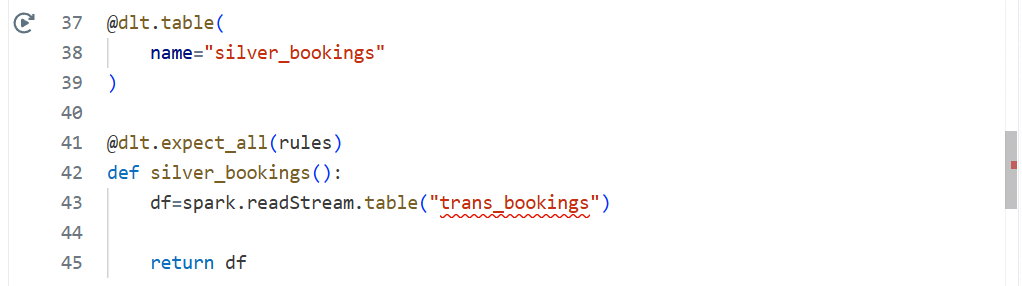
1. Streaming Silver Tables

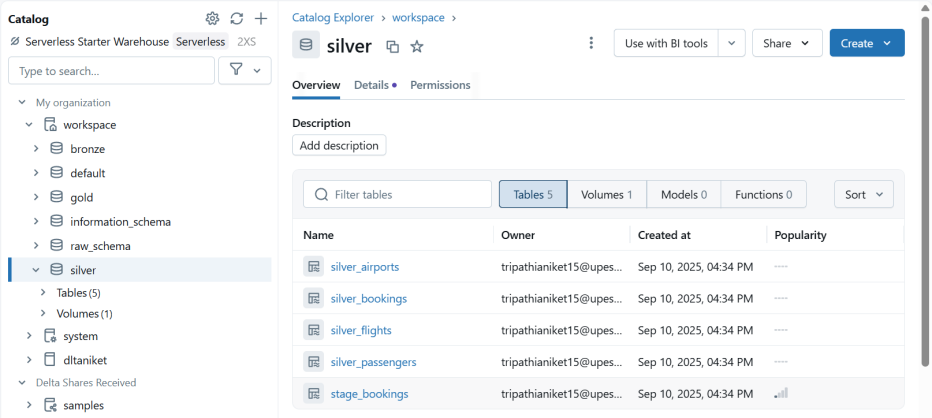
* Defined streaming tables (silver\_bookings, silver\_flights, silver\_passengers, silver\_airports) to continuously update from the Bronze layer
* Ensured that all Silver tables consistently remained up-to-date through incremental data ingestion sourced from the Bronze layer utilizing Autoloader

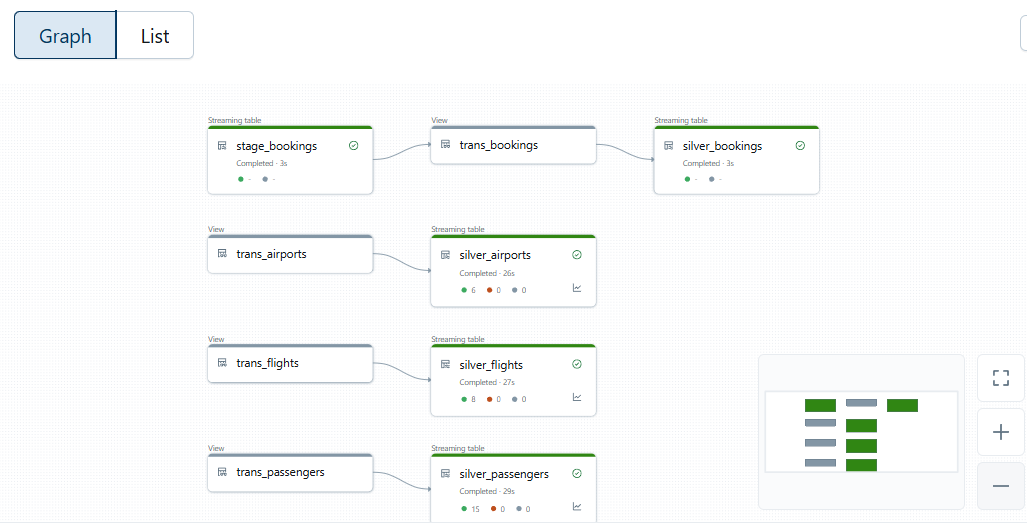
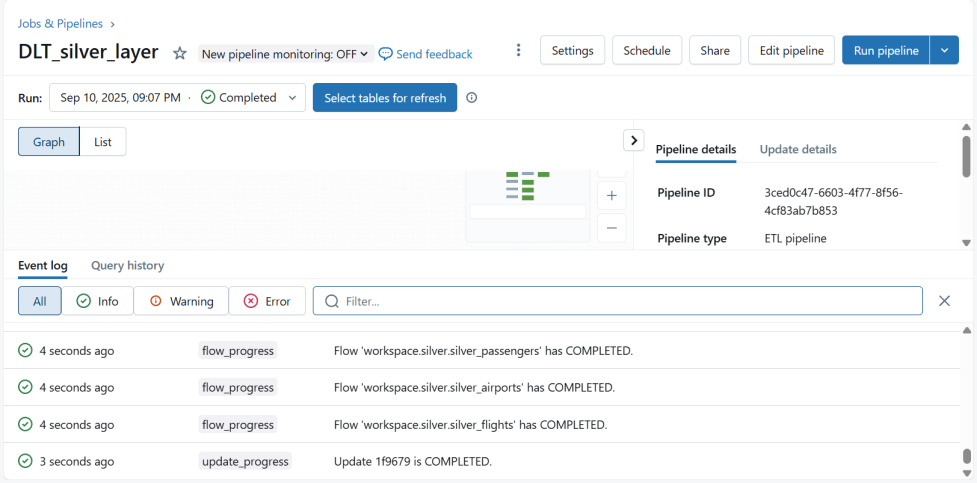
**Code Screenshot for creating silver\_bookings:**











**4. Dimensional Modeling (Gold Layer)**

The Gold Layer was implemented to serve analytics-ready data in a structured, business-friendly format.

Key Activities Performed:

1. Incremental Data Ingestion:

* Each Gold layer table was designed to load data incrementally.
* The pipeline computed the last\_load\_date from existing Gold tables and ingested only records from Silver layer that had a modified\_date more recent than this value.
* This reduced processing overhead and helped refresh large datasets efficiently.

1. Creation of Dimension Tables:

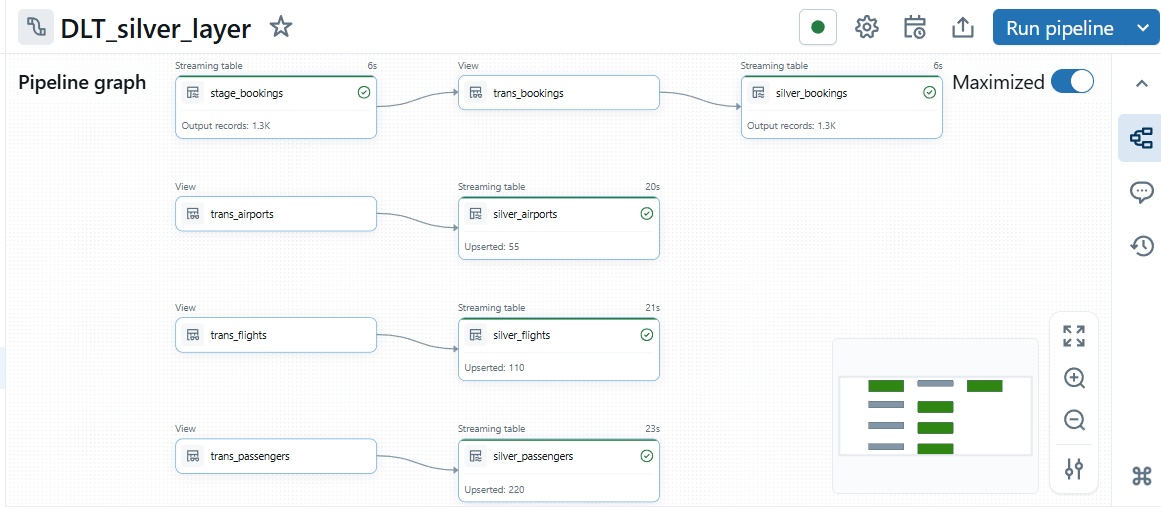
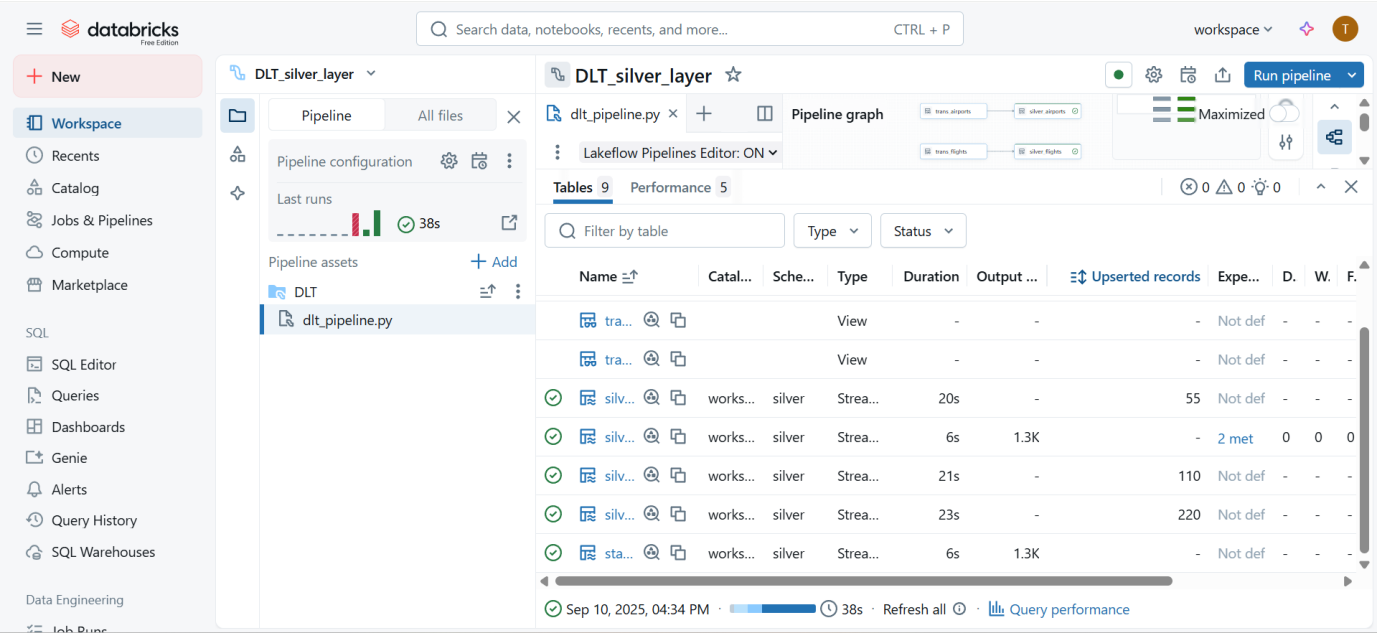
* Dimensions for Passengers (DimPassengers), Flights (DimFlights), and Airports (DimAirports) were created.
* Every dimension employed a surrogate key (DimPassengersKey, for example) to ensure referential integrity.
* The pipeline matched source (Silver) and target (Gold) tables to determine existing and new records:
* Existing records were updated with a new update\_date.
* New records were given surrogate keys and enriched with create\_date and update\_date.
* Applied Slowly Changing Dimensions(Type 1) using Delta Merge(UPSERT) to preserve data consistency while replacing outdated values with latest updates.

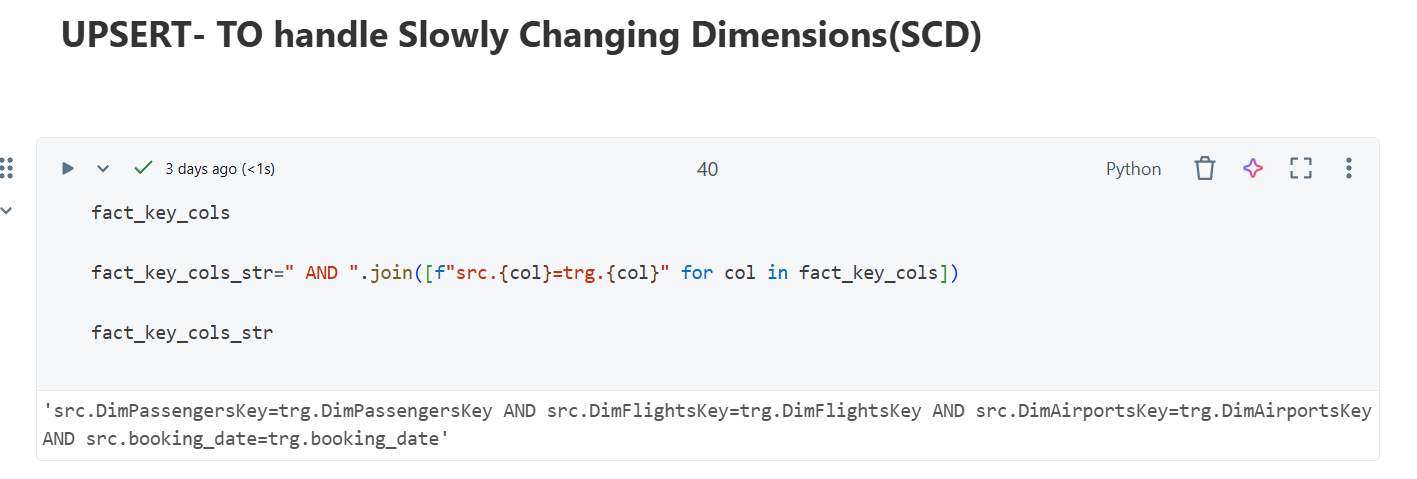
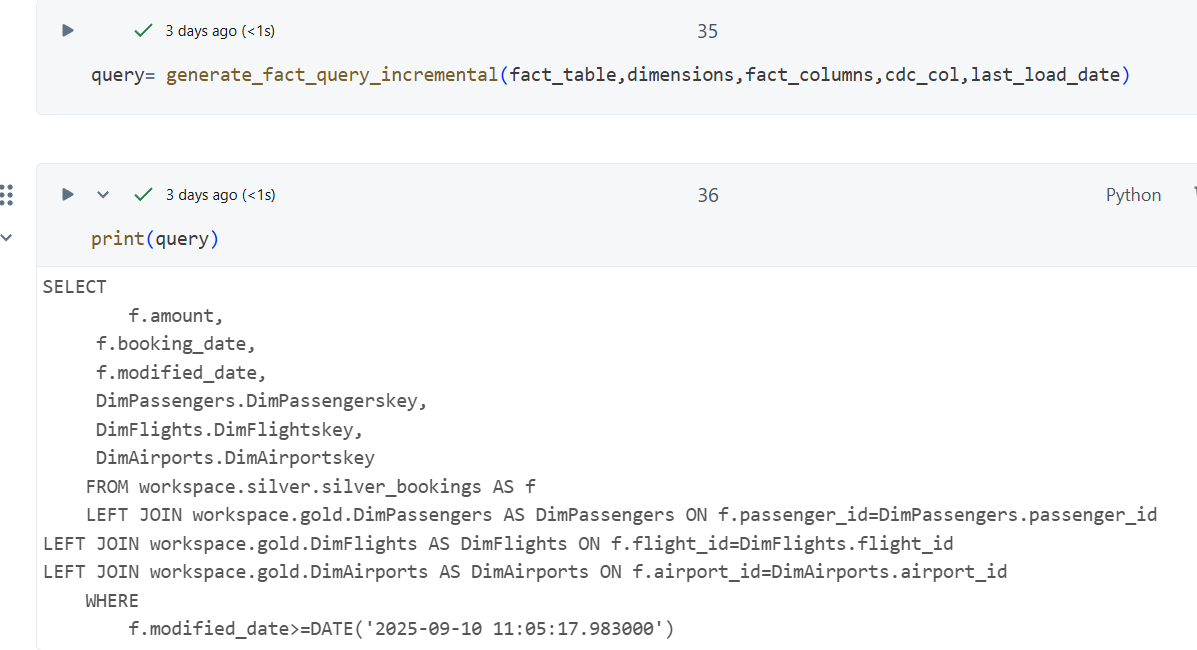
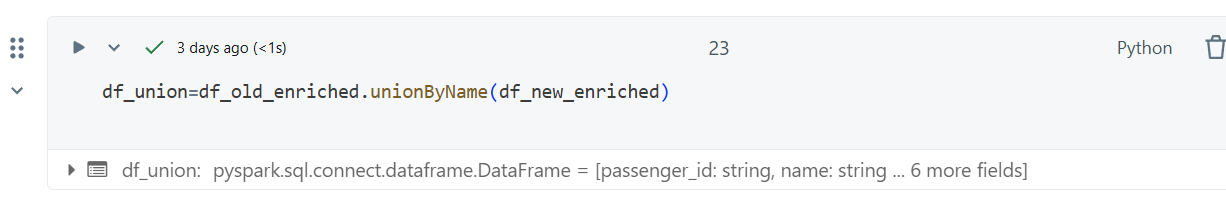
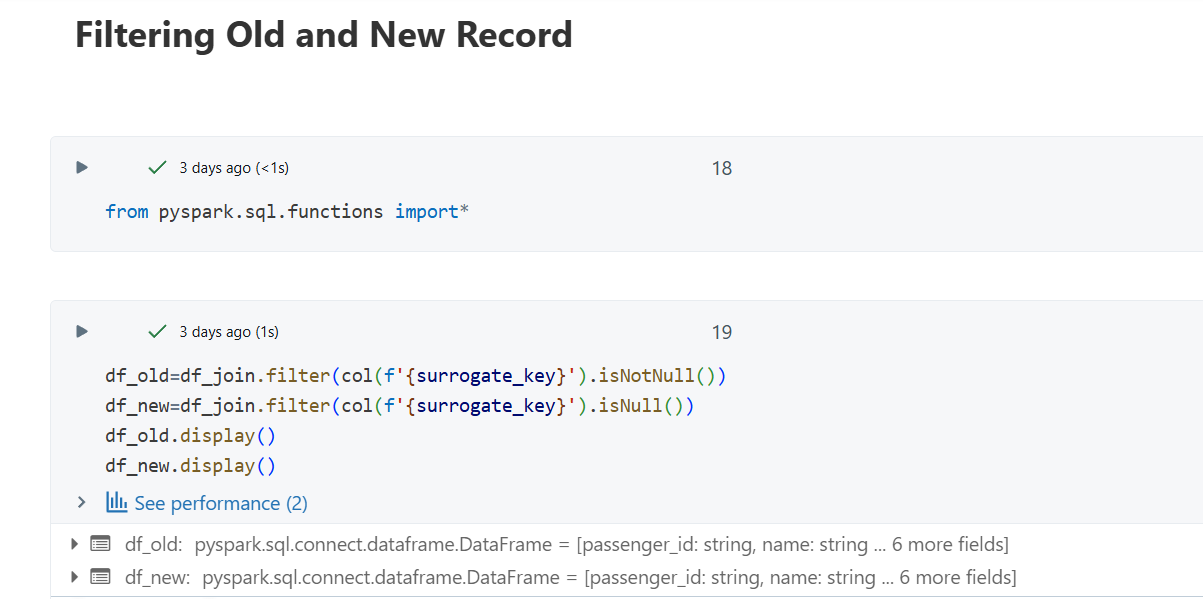
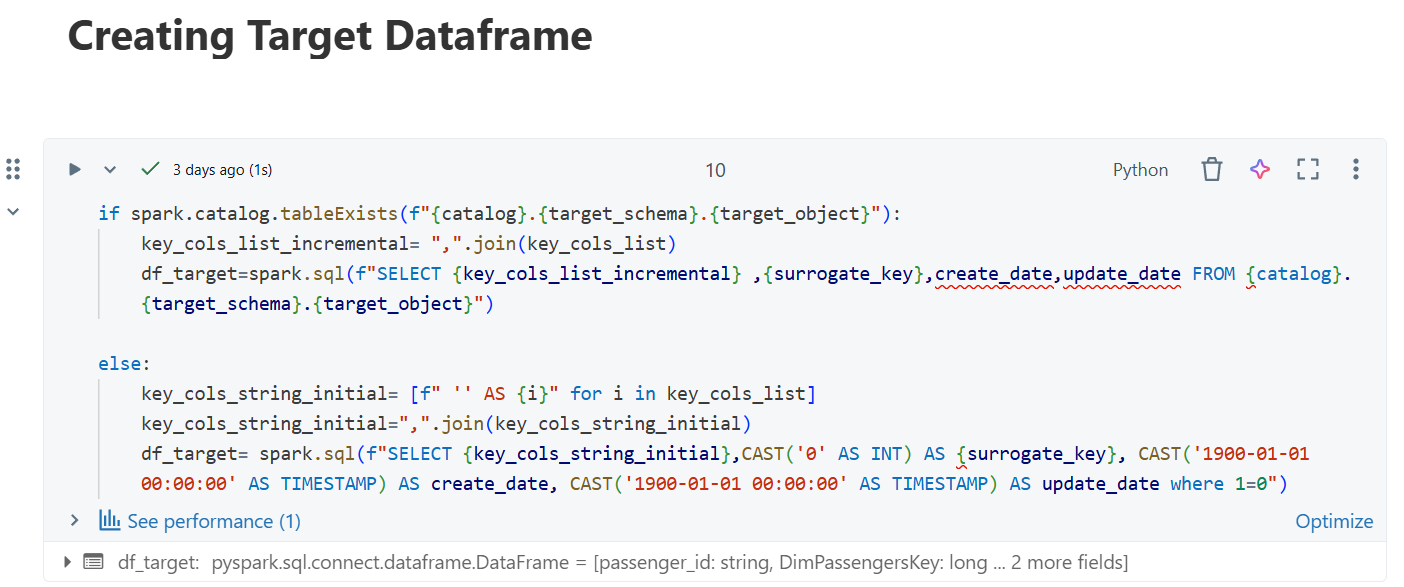
1. Creation of Fact Table:

* A fact\_bookings table was created by joining Silver layer bookings with Gold dimensions
* The pipeline used dynamically generated SQL queries to join the fact table with dimensions on business keys (for instance, passenger\_id, flight\_id, airport\_id).
* The fact table included measures such as booking amount, booking date, modified date, along with surrogate keys from the dimensions for efficient analytical queries.

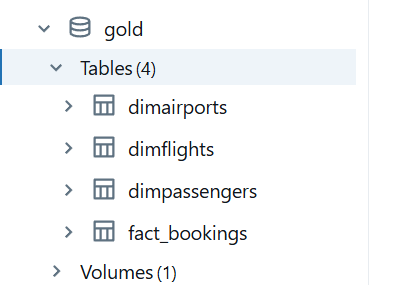
1. Handling Data Changes with UPSERT Logic:

* Delta Lake's MERGE INTO feature was utilized to upsert both dimension and fact table records.
* Ensured that updates (based on modified\_date) were applied to existing records while new inserts were added seamlessly.
* This ensured historical consistency and prevented duplicate records.



****

****

****

## CHAPTER 4

# HARDWARE TOOLS AND COMPONENT

This chapter outlines the hardware tools, system requirements, and cloud components used to design and implement this Project.

* 1. **CLOUD INFRASTRUCTURE:**

The project was deployed on the Databricks Lakehouse Platform which provides an elastic, cloud-based compute environment. Key Infrastructure Features:

* Scalable clusters for distributed data processing
* Delta Lake storage layer for ACID transactions
* Integration with object storage like Azure Data Lake, AWS S3, or DBFS for storing Bronze, Silver, and Gold data.
  1. **Databricks Cluster Configuration:**

The Databrick’s compute cluster was the compute backbone of this project. Databricks Serverless Compute was used, which eliminates the need to manage clusters manually.

**4.3 Hardware Tools and Requirements:**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Component** | **Specification** |
| 1 | Processor(CPU) | Intel Core i5/i7 (8th Gen or higher) or equivalent AMD processor, 2.4 GHz or above |
| 2 | Memory(RAM) | 16 GB DDR4 RAM (Minimum 8 GB for small datasets) |
| 3 | Storage | 512 GB SSD (Minimum 1 TB recommended for large datasets) |
| 4 | Operating System(OS) | Windows 10/11 Pro or Windows Server 2019 |
| 5 | Graphics | Integrated Graphics (No dedicated GPU required) |
| 6 | Network | High-speed Internet (100 Mbps or higher) |

This hardware setup provided a stable and efficient environment for designing, implementing, and testing the data warehouse solution. It also ensures that the project can be deployed to enterprise environments with minimal adjustments.

## CHAPTER 5

# SOFTWARE TOOLS AND LANGUAGES

# The successful execution of the project relied on a set of modern cloud-native software tools and big data technologies. Each tool and programming language was specifically chosen with the aim of efficiency, scalability, and best practices.

**5.1 SOFTWARE TOOLS USED**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **Tool/Software** | **Purpose** | **Remarks** |
| 1 | Databricks Lakehouse | Unified cloud-native platform for big data processing and analytics. | Provided notebooks, storage, and orchestration under a single environment. |
| 2 | Databricks Autoloader | Incremental ingestion of raw files into Bronze layer | Automatically detected new files, handled schema inference and evolution. |
| 3 | Lakeflow Declarative Pipelines(formerly Delta Live Tables) | Declarative pipeline framework for Silver layer transformations. | Applied data quality checks, removed unwanted columns, casted data types, added modified\_date, and supported CDC |
| 4 | Delta Lake | Reliable storage layer with ACID compliance and schema enforcement. | Enabled time-travel queries, MERGE operations for UPSERTs |
| 5 | dbt (Data Build Tool) | Creation of Gold layer fact and dimension tables using modular SQL. | Implemented star schema, incremental models, surrogate keys, and schema tests. |
| 6 | PySpark | Distributed data processing using Spark with Python | Used for transformations, joins, and incremental merges across Silver and Gold layers |
| 7 | GitHub | Version control system for code and pipeline management. | Maintained dbt models, Python scripts, and notebooks |
| 8 | SQL | Primary language for data modeling, data manipulation, and analytics queries | Core language used to define fact/dimension tables and transformations |

# CHAPTER 6

# IMPLEMENTATION

This project was executed on Databricks Lakehouse Platform using serverless compute, Autoloader, Delta Live Tables (DLT), dbt, PySpark, and SQL. The delivery followed a layered approach (Bronze → Silver → Gold) based on the Medallion Architecture.

**6.1 Bronze Layer: Data Ingestion**

* Introduced Dynamic Notebooks to parameterize ingestion logic for several datasets: airports, flights, passengers, and bookings
* Used Databricks Autoloader for incremental loading from cloud storage into Bronze tables.
* Autoloader handled schema inference and evolution, automatically adapting to newly added fields in input data.
* Data were stored in Delta format using partitioning techniques (e.g., based on flight\_date, airport\_code) to boost query performance.

**6.2 Silver Layer: Data Transformation**

* Built a DLT pipeline in Python (dlt\_pipeline.py) that transforms Bronze data into normalized Silver datasets
* Transformation Performed:
* Dropped rescued\_data column
* Casted data types and applied to\_date() for date fields.
* Added a modified\_date field for auditing purposes
* Created DLT views and streaming tables for each dataset (flights, passengers, airports, bookings)
* Applied data quality checks (@dlt.expect\_all) for null constraints and primary keys
* Applied Change Data Capture CDC through dlt.create\_auto\_cdc\_flow in order to deal with updates in flights, airports, and passengers data.

**6.3 Gold Layer: Dimensional Modeling:**

* Developed star schema models for analytical work:
* Dimensions: dim\_flights, dim\_passengers, dim\_airports
* Fact Table: fact\_bookings
* Implemented using dbt, pyspark and SQL:
* dbt modularized SQL transformations
* PySpark handled incremental merges (MERGE INTO) for upserts
* Surrogate keys were introduced for dimension tables
* Incremental refresh implemented by comparing modified\_date between Silver and Gold tables
* Fact table linked bookings with flight, passenger, and airport dimensions

**6.4 Orchestration and Workflow:**

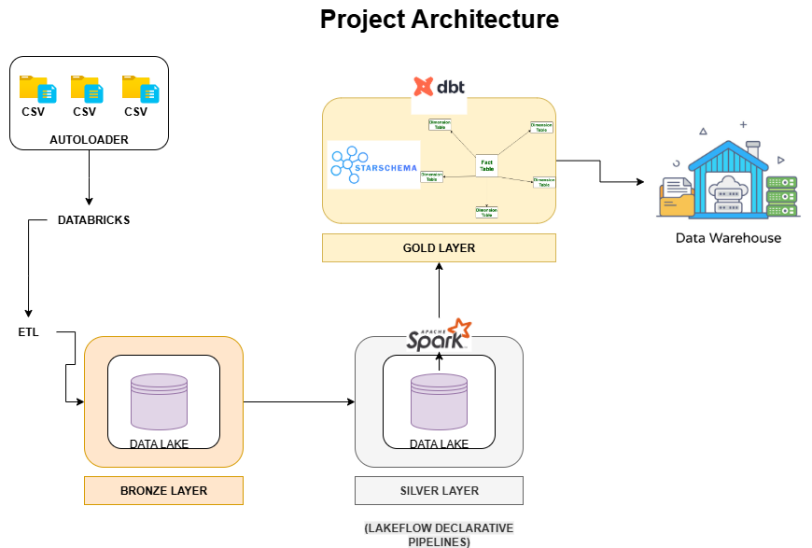
* Bronze layer ingestion pipelines executed via serverless notebooks
* Silver layer transformations orchestrated using the DLT pipeline
* Gold layer executed using Delta Lake MERGE for incremental refresh

A Star Schema was designed for the analytics layer.

* Fact Table(gold.fact\_sales): Stores transactional data like sales\_amount, quantity,price
* Dimesion Tables(gold.dim\_products, gold.dim\_customers): Contains descriptive data like product\_name, first\_name,last\_name
* The schema design simplifies reporting queries and ensures fast aggregations.

**6.5 Outputs:**

* Created analytics-ready cleaned datasets stored in the Gold layer
* Enabled dashboards for:
* Flight Occupancy and Delays
* Passenger Booking Trends
* Airport TrafficAnalysis



# CHAPTER 7

**RESULTS AND DISCUSSIONS**

The implementation of the project produced successful results across all stages of the data pipeline and the outputs from all the three layers i.e. bronze, silver and gold were validated to confirm accuracy, reliability, and readiness for analytics.

|  |  |  |  |
| --- | --- | --- | --- |
| **RESULTS** | | | |
| **Bronze Layer Result** | **Silver Layer Result** | **Gold Layer Result** | **Overall Pipeline Performance** |
| Bronze layer was successfully able to ingest raw data from various sources such as airports, flights, passengers, and bookings together by using Autoloader | The data was cleansed, standardized, and then transformed into well-formed Silver tables in Delta Live Tables (DLT) | Analytics-Ready Star Schema was successfully created  Untitled-Diagram-521 | Incremental loading reduced processing time significantly by avoiding full reloads |
| The ingestion was incremental, ensuring that only new files were loaded | Data Quality Expectations applied were able to enforce type validations, null checks, and a unique primary key | Surrogate keys were successfully created and used for dimensions table, giving referential integrity against the fact table | Serverless compute achieved elastic scaling, which managed big data effectively without requiring manual tuning |
| Schema inference and evolution were automatically handled, reducing the chance of ingestion failures due to schema drift | Incorporation of Change Data Capture (CDC) mechanisms enabled near real-time mirroring of changes in flights, passengers, and airports | UPSERTs achieved through Delta Lake MERGE INTO, ensures successful implementation of SCD-1 | DLT monitoring dashboards provided transparent visibility into pipeline health and data quality problems |
| The raw Delta tables kept data in its original form to maintain auditability | Fact table supplied aggregated booking data along with flight, passenger, and airport links. | Optimization of storage by Delta Lake boosted downstream analytical query performance |

# CHAPTER 8

**CONCLUSION AND FUTURE SCOPE**

**8.1 Conclusion**

This project successfully showcased an end-to-end modern data engineering pipeline based on Databricks Lakehouse Platform, including Serverless Compute, Autoloader, Delta Live Tables (DLT), PySpark, and SQL. The pipeline was executed by utilizing aviation-themed datasets of airports, flights, passengers, and bookings, in accordance with Medallion Architecture principles and thus demonstrated that large, diverse datasets can be ingested, transformed, and modeled in a way that is both scalable and production-ready.

The significance of this work is not simply in building a functional pipeline, but also on bringing together incremental processing, data quality enforcement, and dimensional modeling to provide an ecosystem that effectively supports critical business intelligence in airline industry. The potential to view patterns in bookings, traveler behavior, and airport capacity in near real time represents a fundamental value of this strategy in terms of enhanced decision-makinng and business optimization.

**8.2 Future Scope**

Though the project did develop the initial pipeline successfully, a few improvements could be introduced to expand its scope:

* Real-Time Streaming Integration
* Utilize Kafka or Event Hubs to consume real-time flight and reservation activities.
* Provide real-time dashboards for operations observation (e.g., current flight delay, current passenger check-ins).
* Machine Learning and Advanced Analytics:
* Using Databricks ML feature, create predictive models for:

1. Passenger demand prediction
2. Dynamic pricing optimization
3. Flight delay prediction

* Data Governance and Security:
* Use Unity Catalog for granular data access controls and lineage tracking
* Incorporate role-based access to datasets by various stakeholders (analytics teams, finance, operations)
* Scalability and Multi-Cloud Integration:
* Expand ingestion to several regions and airlines.
* Allow cross-cloud interoperability (AWS, Azure, GCP) for worldwide deployment.
* Visualization and BI Dashboards
* Support integration with Tableau or PowerBI to create interactive advanced dashboards.
* Provide executives with real-time KPIs such as passenger load factor, revenue per flight, and airport traffic heatmaps

**REFERENCES**

1. Databricks. (2022). The Medallion Architecture in the Lakehouse Design. Databricks Blog. Available at: https://databricks.com/blog
2. Databricks. (2023). Delta Live Tables Documentation. Available at: https://docs.databricks.com/delta-live-tables
3. Databricks. (2023). Autoloader Documentation. Available at: https://docs.databricks.com/ingestion/auto-loader
4. Databricks. (2023). Delta Lake Documentation. Available at: https://docs.databricks.com/delta
5. Inmon, W. H. (2005). Building the Data Warehouse (4th ed.). Wiley Publishing.
6. Kimball, R., & Ross, M. (2013). The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling (3rd ed.). Wiley Publishing.
7. Apache Software Foundation. (2023). Apache Spark Documentation. Available at: https://spark.apache.org/docs
8. Ansh Lamba. (2025). Databricks x DBT End-to-End Data Engineering Project [Job Ready 2025] [Video]. YouTube. Available at: [https://www.youtube.com/](https://www.youtube.com/" \t "_new)