✈️ Flights Booking Pipeline – Functional Requirements & Documentation

1. Business Understanding

The Flights Booking Pipeline supports real-time and batch analytics for aviation datasets—**bookings, flights, passengers, and airports**—using the Databricks Lakehouse Medallion Architecture:

- **Raw Layer**: CSV file landing zone
- **Bronze Layer**: Delta ingestion (with schema evolution)
- **Silver Layer**: DLT streaming, CDC & business transformation
- **Gold Layer**: Star schema modeling, SCD, Fact/Dim separation

Business outcomes:

- Operational dashboards (trends, availability)
- Customer analytics
- Predictive ML (demand, churn)

2. Project Plan

Phase	Description
Step 1	Raw & Bronze: Autoloader for each domain's CSV + schema evolution
Step 2	Silver Layer (DLT): Type casting, business rules, validation
Step 3	Gold Layer: Star schema build, surrogate keys, SCD/metrics
Step 4	Dashboarding: dbt, Power BI, and ML consumption

3. Initial Data Collection Report

Sources: CSVs in Unity Catalog Volumes.

Format: Csv with headers, schema differs by domain.

Ingestion: Databricks Auto Loader, schema rescue mode.

Domain	File Name	Target Volume Path
Bookings	bookings.csv	/Volumes/workspace/raw/rawvolume/rawdata/bookings
Flights	flights.csv	/Volumes/workspace/raw/rawvolume/rawdata/flights
Passengers	passengers.csv	/Volumes/workspace/raw/rawvolume/rawdata/customers
Airports	airports.csv	/Volumes/workspace/raw/rawvolume/rawdata/airports

4. Data Description Report

Domain	CSV Columns
Bookings	booking_id, passenger_id, flight_id, airport_id, amount, booking_date
Flights	flight_id, airline, origin, destination, flight_date
Passengers	passenger_id, name, gender, nationality
Airports	airport_id, airport_name, city, country

5. Data Quality Report

- **DLT Data Quality:**
  - `booking_id IS NOT NULL`, `passenger_id IS NOT NULL`
  - Drop malformed records via `_rescued_data`
- **Schema evolution:**
  - Bronze: `.option("cloudFiles.schemaEvolutionMode", "rescue")`

6. Data Selection Report

🎯 Gold Layer Dimensions (Surrogate Keys & Attributes):

Domain	Natural Key	Surrogate Key	Attributes
Passengers	passenger_id	DimPassengersKey	name, gender, nationality, create_date, update_date
Flights	flight_id	DimFlightsKey	airline, origin, destination, flight_date, create_date, update_date
Airports	airport_id	DimAirportsKey	airport_name, city, country, create_date, update_date

📊 Gold Layer Fact Table

FactBookings

Column Name	Description
booking_id	Transaction/business key (for trace/audit)
DimPassengersKey	FK to DimPassengers
DimFlightsKey	FK to DimFlights
DimAirportsKey	FK to DimAirports
amount	Booking amount
booking_date	Booking event date
modifiedDate	CDC marker (from Silver Layer)

Primary Key: Composite of all Dim*Key FKs + booking_date

7. Data Cleaning Report

- **Silver via DLT**:
    - Type cast: amount → DoubleType, booking_date, flight_date → DateType
    - modifiedDate = current_timestamp()
    - Drop: _rescued_data

8. Data Derivation Report

- **Silver:** add modifiedDate
- **Gold:** add create_date, update_date in all dimension tables

- **Surrogate Key Generation:**
  `max_sk + 1 + monotonically_increasing_id()`
- **Dimension Types:** All are SCD Type 1 (latest only, can test SCD Type 2)

9. Data Modeling Report

★ Star Schema Design

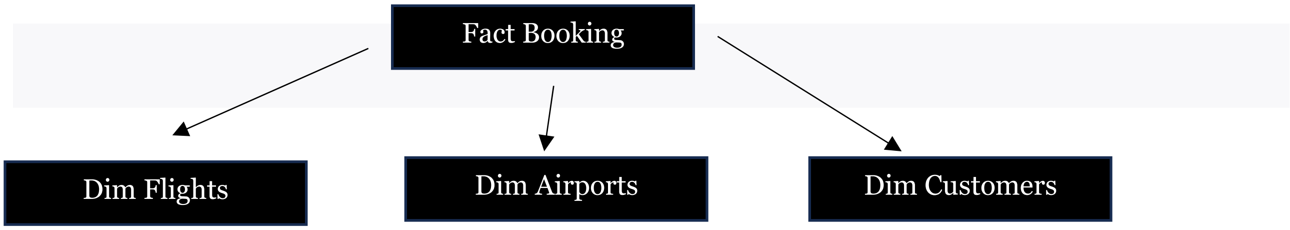


Table Snapshots:

DimFlights

DimFlightsKey	flight_id	airline	origin	destination	flight_date	create_date	update_date

DimAirports

DimAirportsKey	airport_id	airport_name	city	country	create_date	update_date

DimPassengers

DimPassengersKey	passenger_id	name	gender	nationality	create_date	update_date

FactBookings

booking_id	DimPassengersKey	DimFlightsKey	DimAirportsKey	amount	booking_date	modifiedDate

🔁 Checkpointing Summary

Layer	Mechanism Used	Location / Format
Bronze	Spark Structured Streaming + Auto Loader	/Volumes/workspace/bronze/bronzevolume/<domain>/checkpoint
Silver	Delta Live Tables (DLT) Streaming	Managed internally by the DLT pipeline (no manual checkpoint)
Fact/Dim	Surrogate Key Tracking + Last Modified Date	Incremental MERGE/upsert logic using modifiedDate (no file location, tracked by data content)

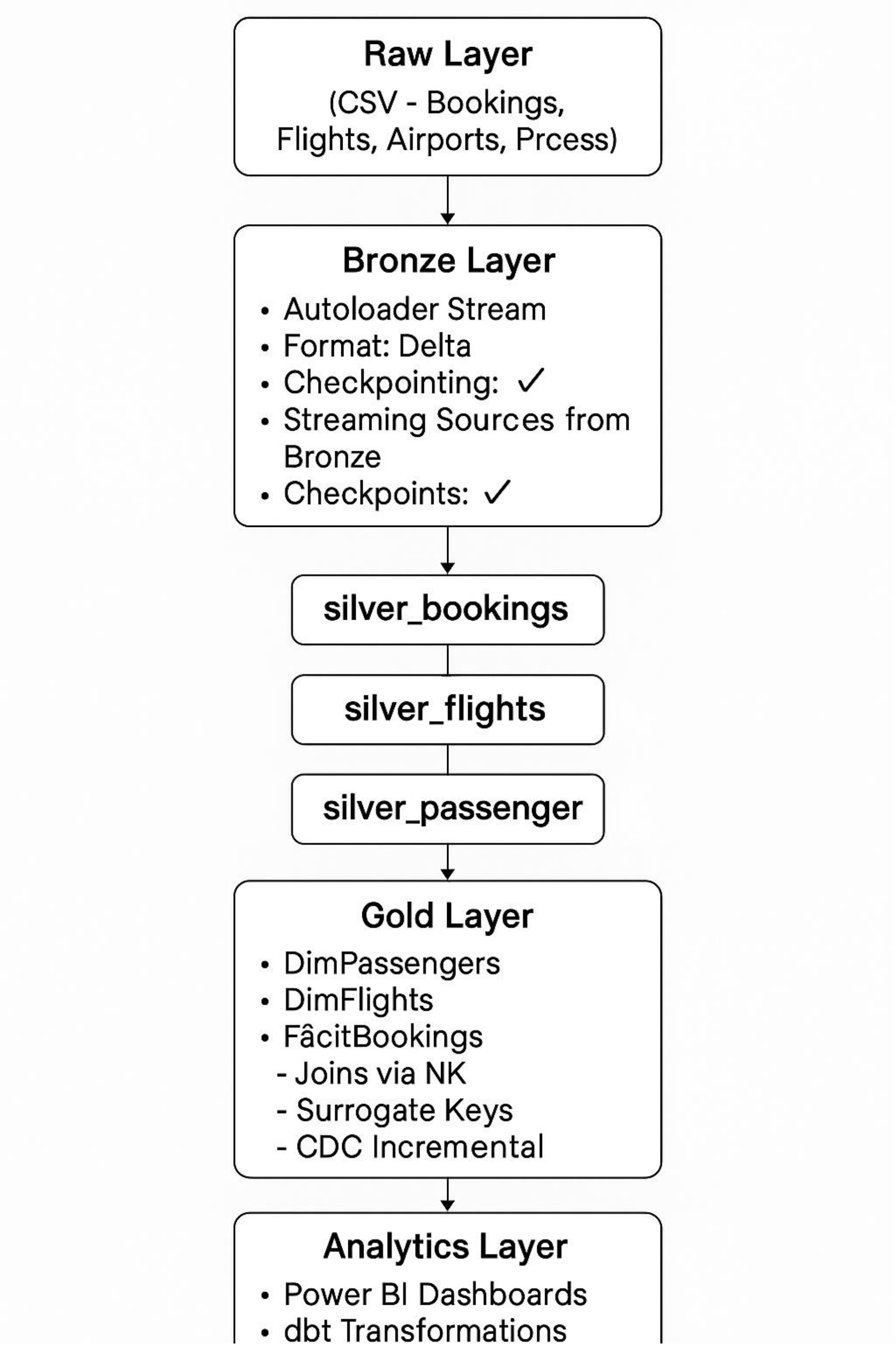
Notes:

- **Bronze:** Checkpoint folders persist ingestion state for idempotent streaming.
- **Silver:** DLT manages checkpoints, tracking streaming/run state automatically.
- **Fact/Dim (Gold):** CDC/incremental logic is based on modifiedDate and surrogate key values, so change tracking is “in-table” rather than filesystem-based.

Final Notes

- All tables built dynamically via parameters (catalog, CDC column, keys).
- Incremental ingestion, upserts, and SCD audit history managed in pipeline.
- Consistent columns and naming from source to analytics layers.

ARCHITECTURE DESIGN


Raw Layer
(CSV - Bookings,
Flights, Airports, Prcess)
Bronze Layer
• Autoloader Stream
• Format: Delta
• Checkpointing: ✓
• Streaming Sources from Bronze
• Checkpoints: ✓
silver_bookings
silver_flights
silver_passenger
Gold Layer
• DimPassengers
• DimFlights
• FâcitBookings
- Joins via NK
- Surrogate Keys
- CDC Incremental
Analytics Layer
• Power BI Dashboards
• dbt Transformations