Airline Data Management and Analysis Using Power BI

Submitted By: - Aman Kumar Yadav

Course: - Data Science Placement Guarantee Course

Submission Date: - 13th July 2025

Table of Contents: -

Introduction

The airline industry generates and relies heavily on large volumes of data, including flight schedules, passenger bookings, ticket status, and operational metrics. Managing and analysing this data effectively can significantly improve decision-making, resource allocation, and customer satisfaction.

In this project, we focus on analysing airline data using Power BI. The goal is to visualize operational patterns and uncover key insights by cleaning, transforming, and modelling datasets containing information about flights, passengers, and tickets.

By creating an interactive and dynamic Power BI dashboard, we aim to help airline operations teams understand trends, monitor performance, and identify areas of improvement.

Objectives of the Project:

- Analyse flight, passenger, and ticket data
- Clean and transform raw data using Power Query
- Build a data model with appropriate relationship
- Create interactive reports and dashboards
- Generate insights for operational improvements

Tools Used:

- Microsoft Power BI
- Power Query Editor
- DAX (Data analysis Expressions)

Data Preparation and Cleaning

In this step, I imported all three datasets: - Flight_Information, Ticket_Information, and Passenger_Information – into Power BI using Power Query Editor.

The Data preparation involved the following actions: -

- **Removed duplicate records** to ensure data integrity.
- **Eliminated blank rows** that could affect data modelling
- **Promoted headers** to convert the first row into column names.
- **Checked column validity** to confirm all values are in proper format (e.g., FlightID is numeric, FlightNumber is text).
- **Previewed data profiling** to ensure completeness and consistency

The following Screenshots shows the cleaned datasets in the Power Query Editor after all transformations were applied: -

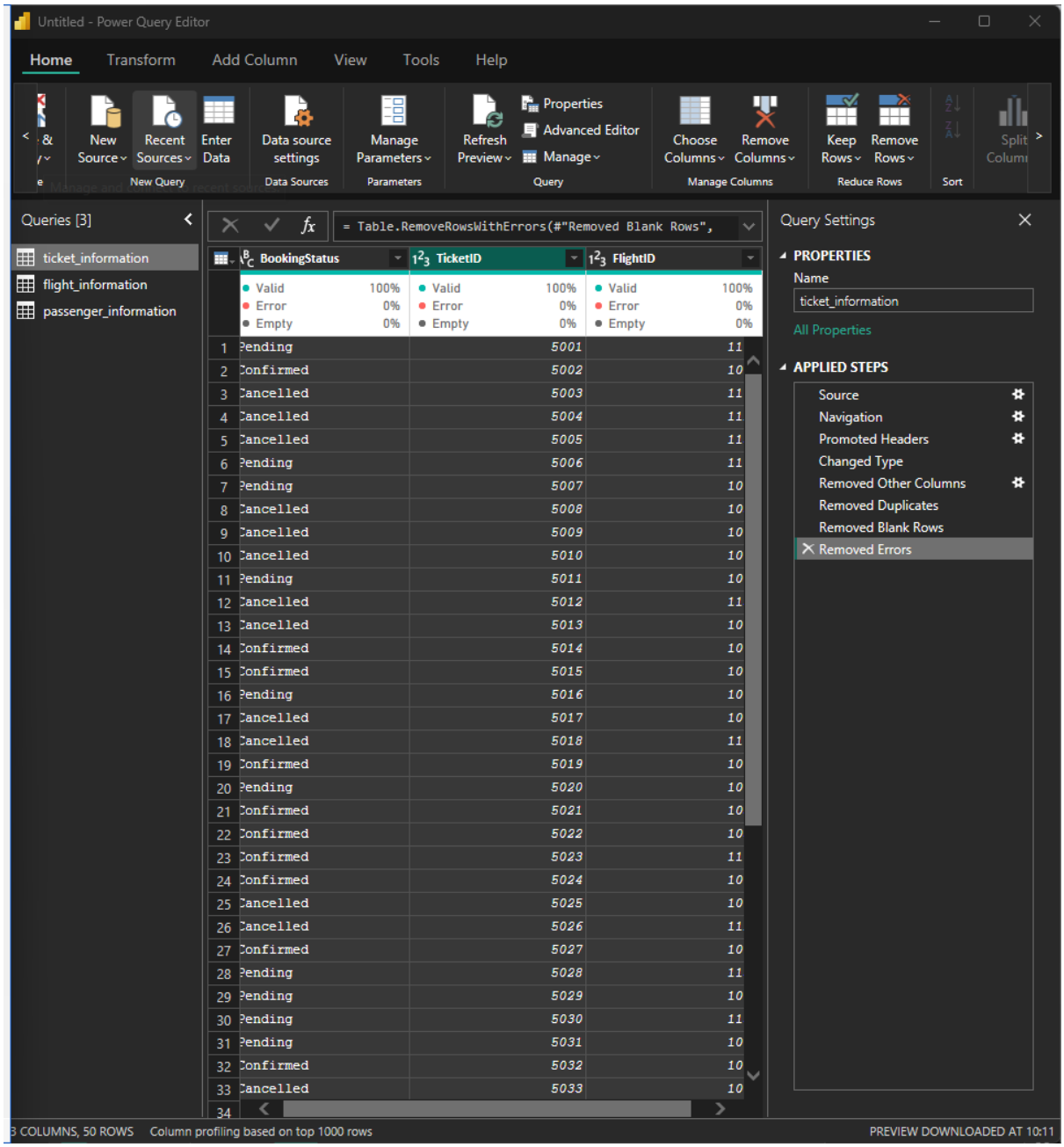


Figure 1: Cleaned Ticket_Information Dataset in Power Query

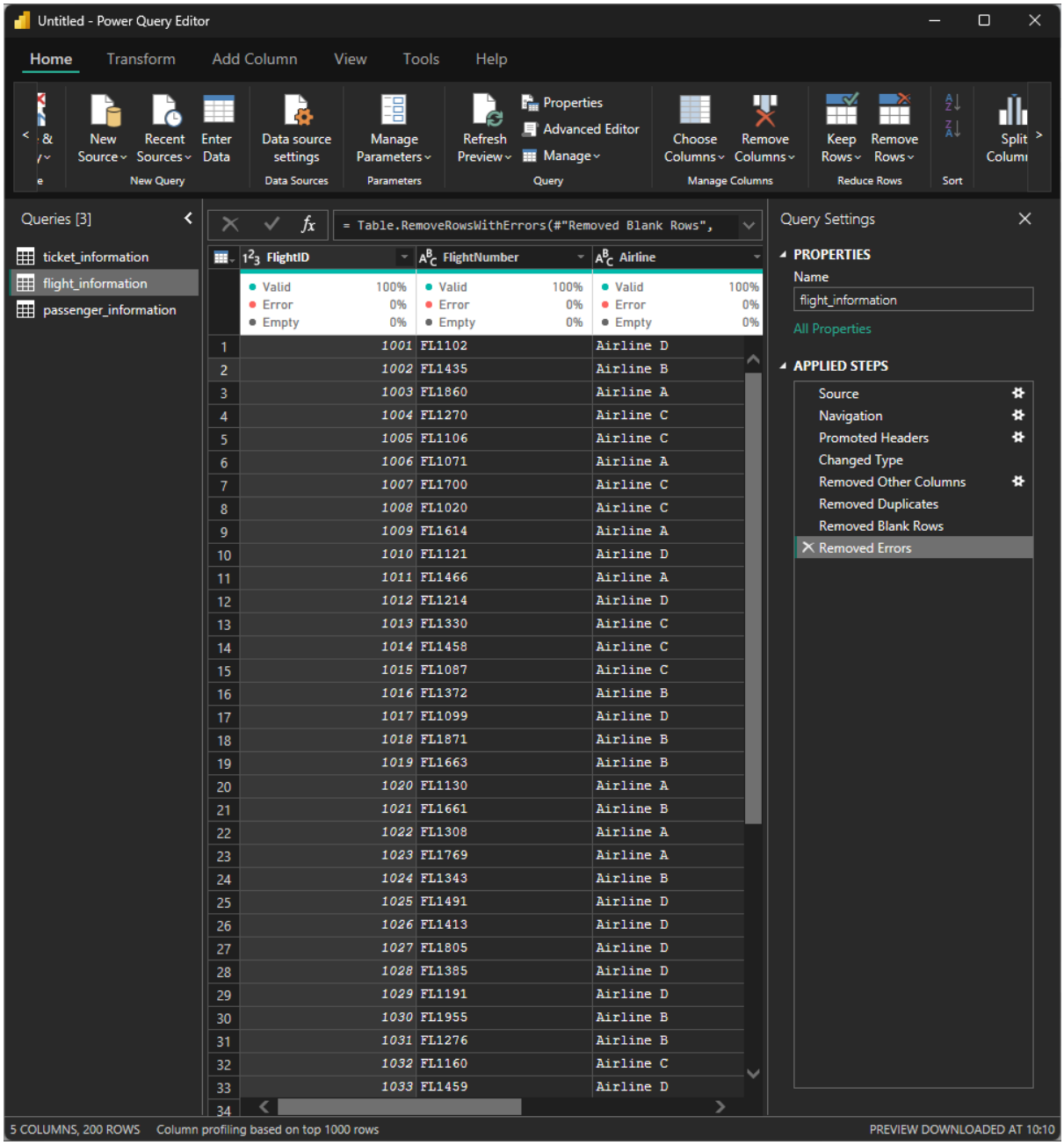


Figure 2: - Cleaned Flight_Information Dataset in Power Query

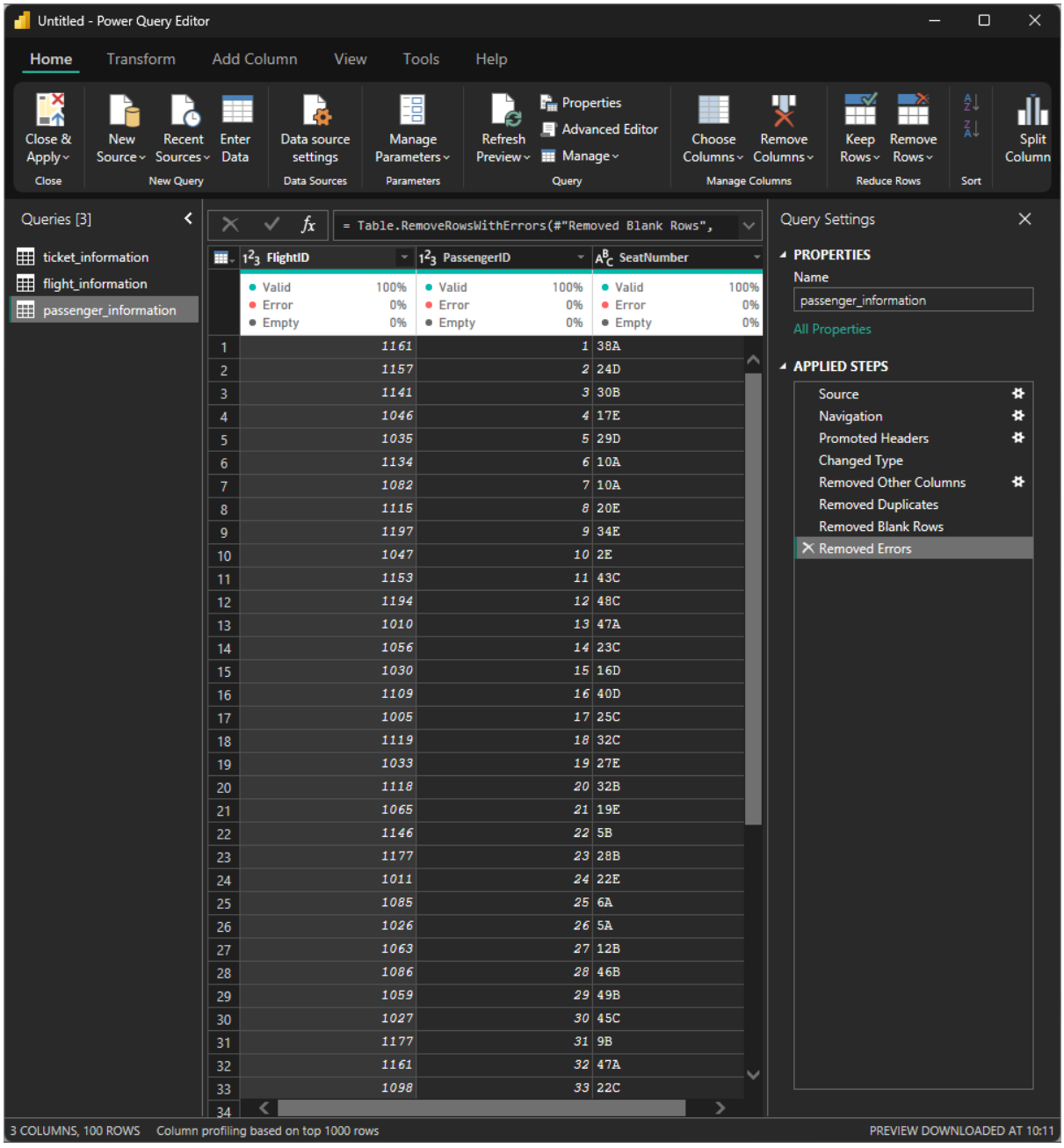


Figure 3: - Cleaned Passenger_Information dataset in Power Query

Data Modelling

After cleaning the data, I moved to the Model View in Power BI to establish relationships among the three datasets: -

1. Flight_Information
2. Passenger_Information
3. Ticket_Infromation

Relationships Created: -

- **Flight_id** is the **Primary key** in the **flight_Information** Table.
- **Passenger_Infromation** and **Ticket_Information** tables both use **FlightID** as a **Foreign key.**
- These relationships follow a one-to-many cardinality, where one flight can have many passengers and tickets.

Cross-filter Direction: s-

- I kept the default single filter direction to maintain a simpler model.
- No circular dependencies were present.

The following Screenshot shows the data model with relationships:-

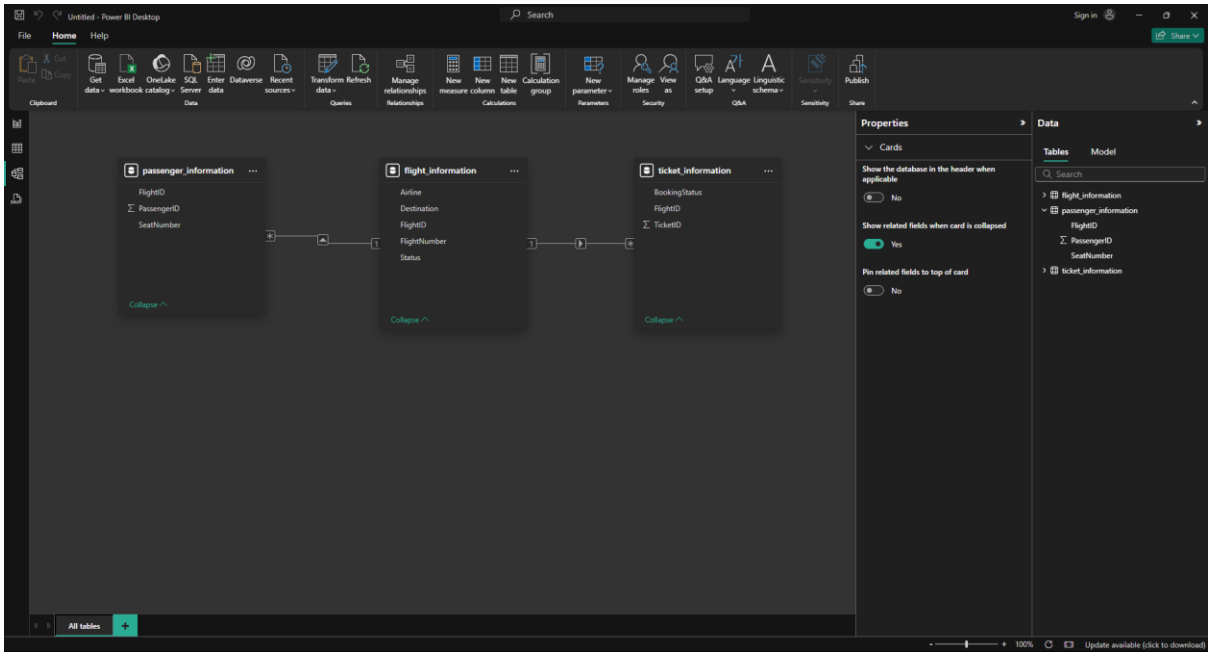


Figure 4: - Data Model Showing Relationships using FlightID

Enhanced Data Insights

To gain better insights from the flight data, I enhanced the dataset with two new columns using Power Query Editor.

1. Conditional Column: - FlightStatusLabel
   I added a new column **FlightStatusLabel** in the **Flight_Information** table using the Conditional Column feature. The column classifies flights as:
   - "Best" – if the flight status is On Time.
   - "To Be Improved" – if the flight status is Delayed or Cancelled

   This allows better filtering and visualization of flight performance.

2. Extracted Flight Number: -
   Using the "Column from Examples", I created a new column **FlightNumberOnly** by extracting the numeric part of the FlightNumber field. For Example:-

   - From FL1102 → Extracted 1102

This helps in sorting, filtering, or grouping flights based on numerice Flight IDs.

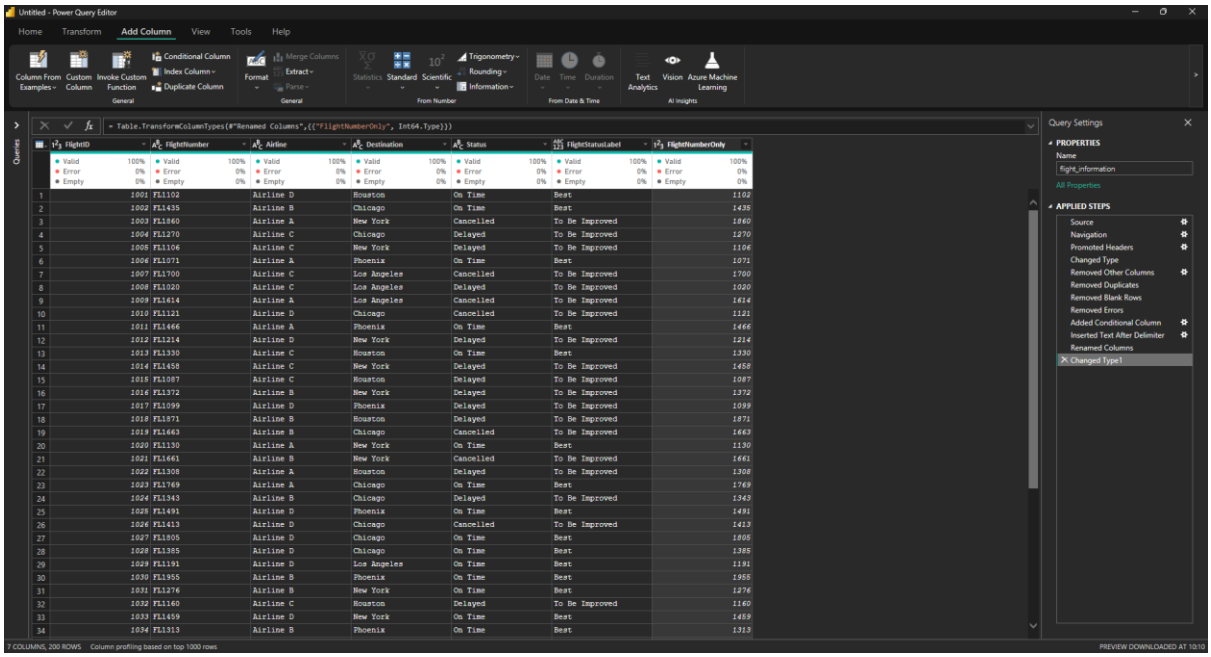


Figure 5: - Enhanced Flight Information with Conditional Column and Extracted Flight Number.

DAX Calculations

1. Total Passengers for a Specific Flight: -

   To analyse the number of passengers on a particular flight, I created a DAX measure using the CALCULATE () and COUNTROWS () functions.

   The goal was to calculate the total number of passengers who booked Flight FL1102 which corresponds to FlightID = 1001 in the dataset.

```
TotalPassengers_FL1102 = CALCULATE(
    COUNTROWS(passenger_information),
    passenger_information[FlightID] = 1001
)
```

Figure 6: - DAX Measure Calculating Total Passengers for FL1102

2. Total Confirmed Tickets: -

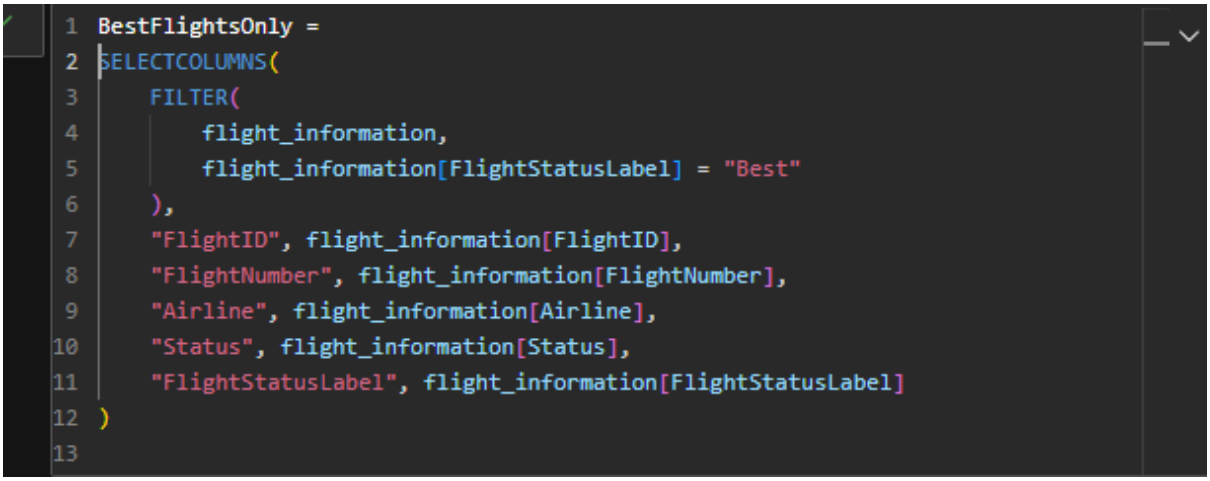
   I created a DAX measure to count only the tickets with a BookingStatus marked as “Confirmed”, to understand the actual number of passengers who successfully booked flights.

```
Total Tickets Booked = CALCULATE(COUNTROWS(ticket_information),
ticket_information[BookingStatus] = "Confirmed")
```

Figure 7: - DAX Measure for Total Confirmed Tickets

3. Filtered Table for Best Flights Only: -

   To focus on well-performing flights, I created a new calculated table using DAX that filters the flight_information table for rows labelled “Best” in the FlightStatusLabel column (which was created in Step 3).



```
BestFlightsOnly =
SELECTCOLUMNS(
    FILTER(
        flight_information,
        flight_information[FlightStatusLabel] = "Best"
    ),
    "FlightID", flight_information[FlightID],
    "FlightNumber", flight_information[FlightNumber],
    "Airline", flight_information[Airline],
    "Status", flight_information[Status],
    "FlightStatusLabel", flight_information[FlightStatusLabel]
)

```

Figure 8: - Best Flights Table Filtered using DAX

Visualisation and Interactive features

1. Passenger Count by Airline: -
   A bar Chart was created to show the total number of passengers served by each airline. This Chart uses the Airline column on the axis and count of passengers using the TotalPassengers measure or row count.

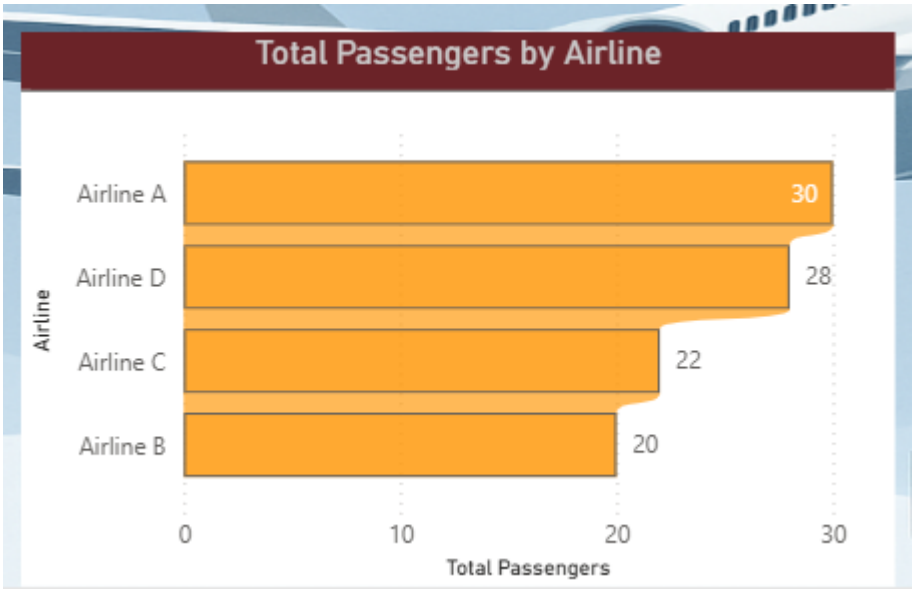


Figure 9: - Bar chart → Passengers Count by Airline

2. Ticket Booking Statuses: →
   A donut chart was used to represent the distribution of different booking statuses (e.g., Confirmed, Cancelled). This gives a quick view of booking outcomes.

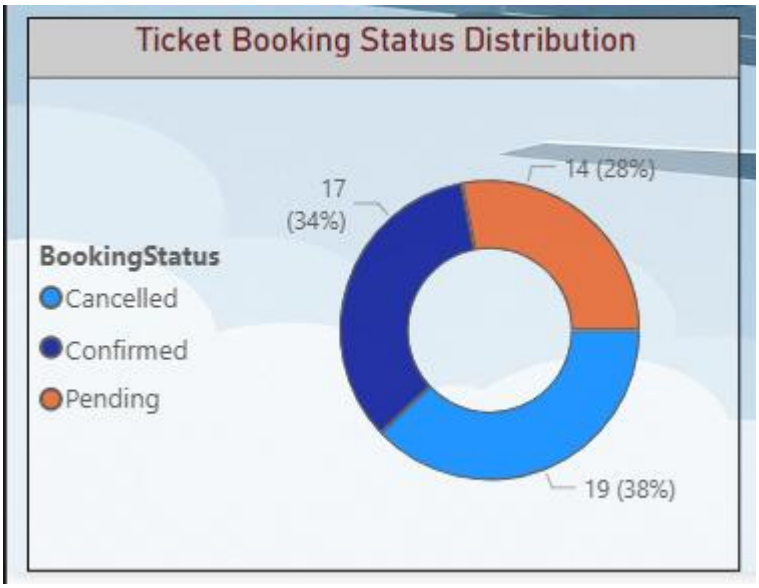
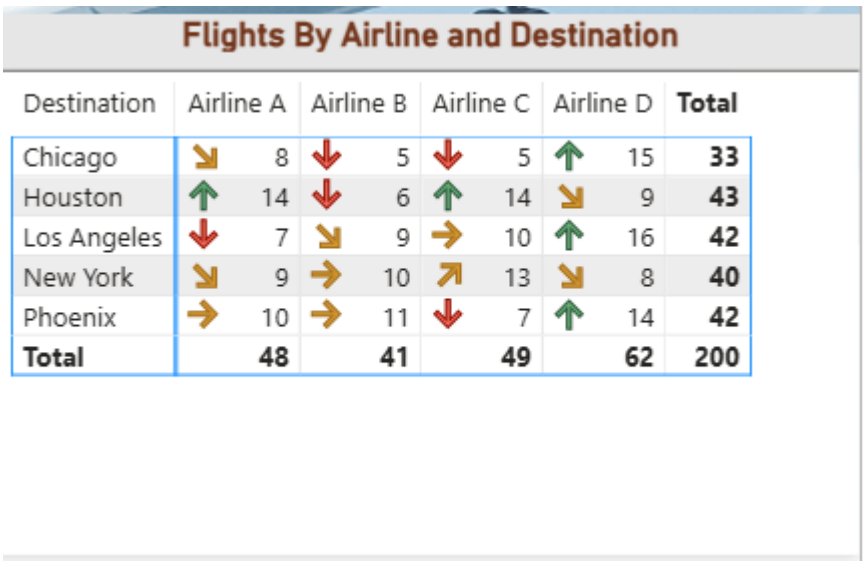


Figure 10:- Donut Chart → Ticket Booking status Distribution

3. Flights By Airline and Destination: →
   A matrix was added to analyse how many flights each airline operates to various destinations.
   - Rows: Destinations
   - Columns: Airline
   - Values: count of FlightID



Flights By Airline and Destination

Destination	Airline A	Airline B	Airline C	Airline D	Total
Chicago	8	5	5	15	**33**
Houston	14	6	14	9	**43**
Los Angeles	7	9	10	16	**42**
New York	9	10	13	8	**40**
Phoenix	10	11	7	14	**42**
Total	**48**	**41**	**49**	**62**	**200**

Figure 11:- Matrix → Flights by Airline and Destinations

4. Slicers Used: →
   - Airline Slicer: - Allows filtering all visuals to show data only for the selected airlines.
   - Destination Slicer: - Filters Visuals to display flights and tickets related to specific destinaitons.

Figure 12: → Interactive Slicers Filtering ALL Dahsboard Visuals

5. Confirmed Tickets by Destination: →
   To analyse demand across different travel routes, I created a clustered column chart that display the number of confirmed tickets for each destination.
   This visual helps identify: -
   - Which destinations are most popular
   - Where the highest booking volume occurs
   - Pattern in customer travel behavior

Figure 13: → Confirmed Tickets by Destinations

Final Dashboard and Power BI Service

The final version of the Power BI dashboard was published to the Power BI Service for online access and sharing. Key deployment steps included:

1. Publishsing:-
   The .pbix file was uploaded to Power BI Services via My Workspace.

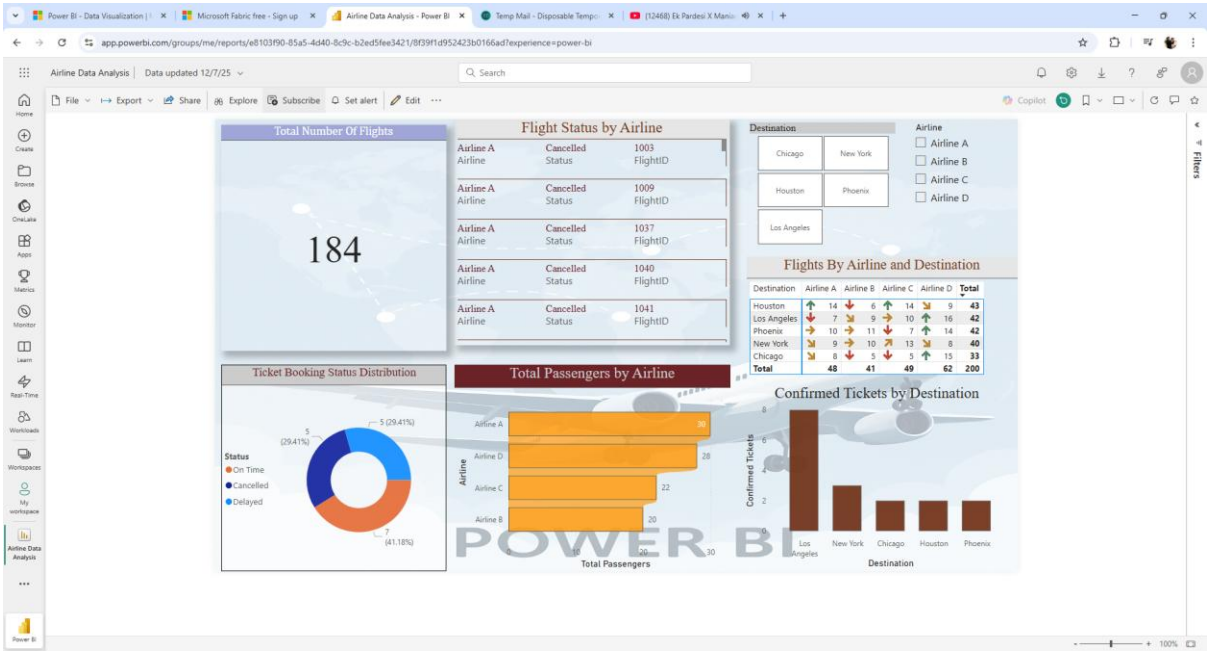


Figure 14: → Published Dashboard on Power BI Sevice

2. Row-Level Security (RLS):→
   To Restrict data visibility for specific users, a Row-Level Security (RLS) role named AirlineARole was created.
   This role limits access to records where the Airline columns equals “Airline A” using the filter expression.

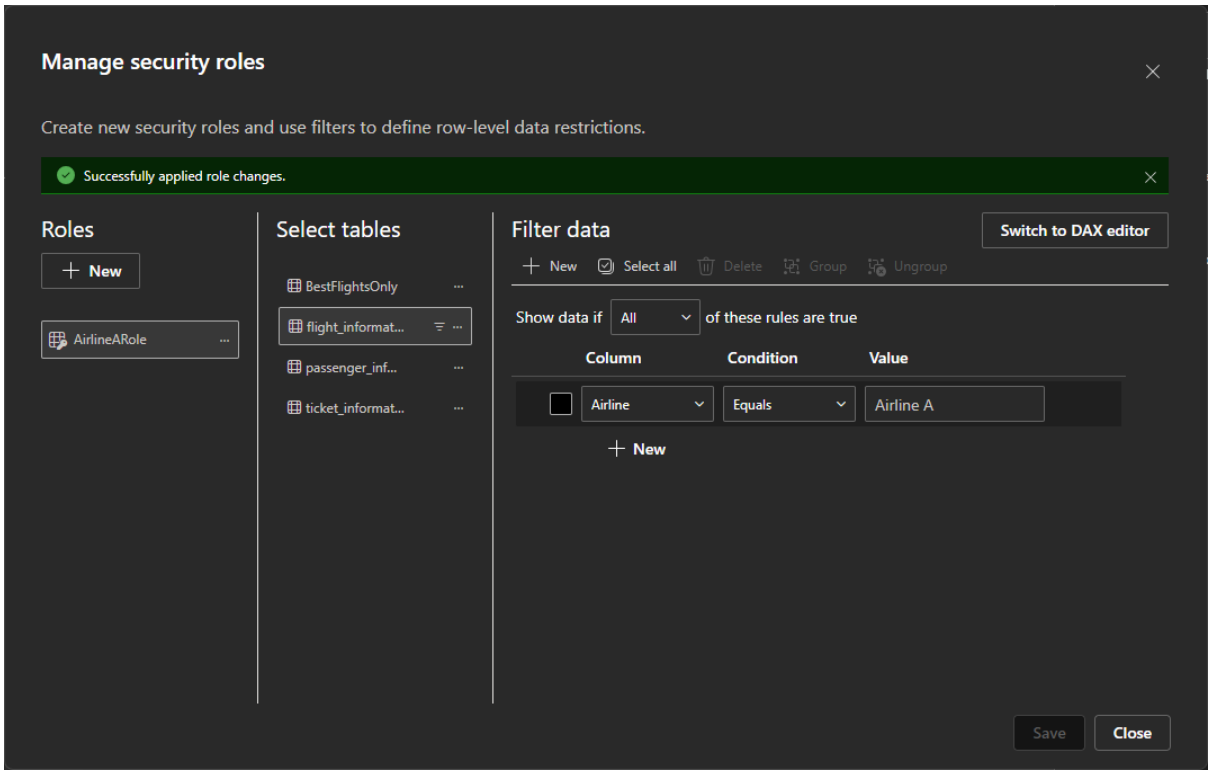


Figure 15: → RLS Role Setup in Power BI desktop for Airline A

3. Testing RLS: →
   After creating the RLS role ArilineARole, the “View as” feature in Power BI desktop was used to simulate the view for users assigned to this role.
   This allowed testing of the restriction Airline = “Airline A” to confirm that only data related to Airline A is Visible when the role is applied.

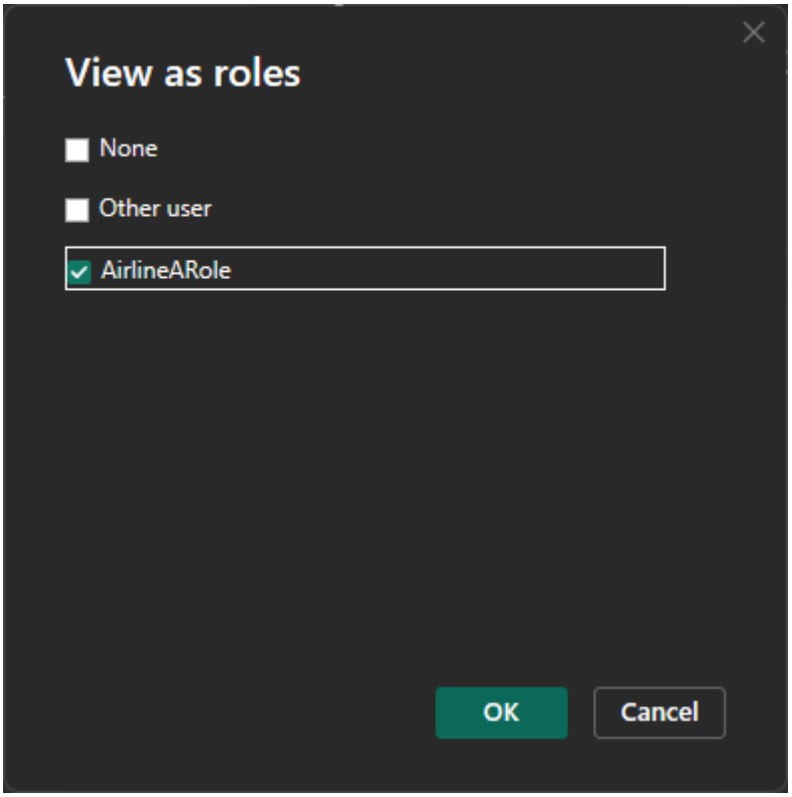


Figure 16:- Testing RLS using "View as role" feature in Power BI Desktop

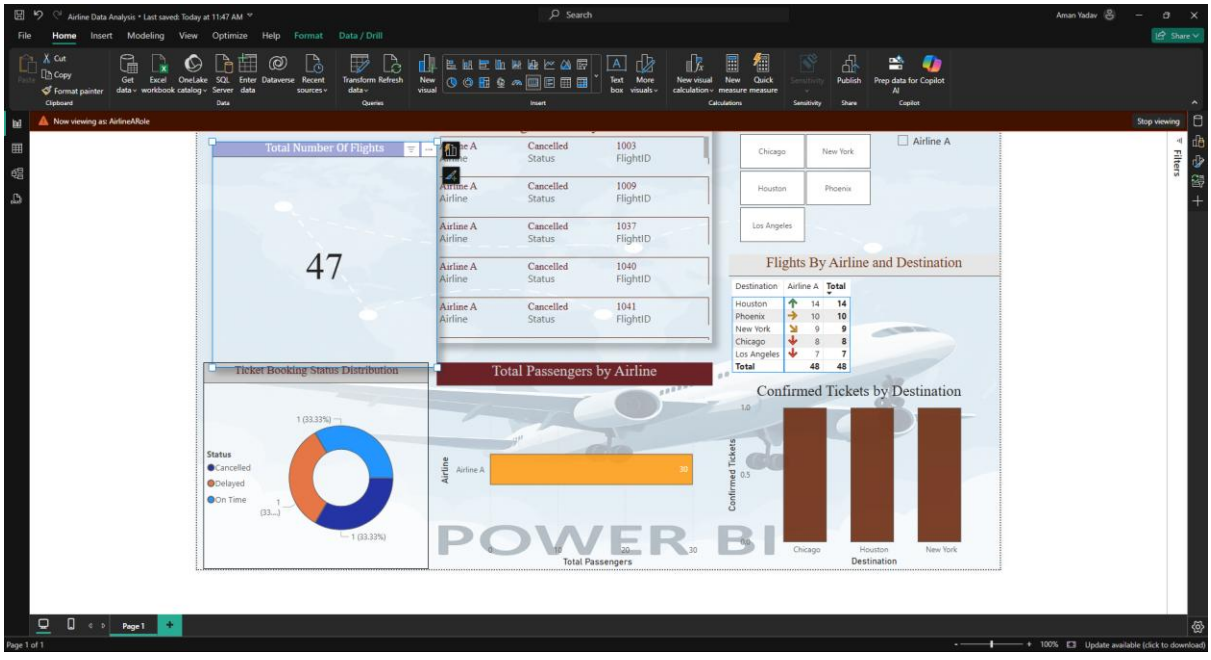


Figure 17:→ Dashboard View with AirlineARole Applied (RLS Active)

4. Scheduled Refresh setup (5 PM): →
   To automate data updates, a daily scheduled refresh was configured in Power BI Service using the default gateway. The refresh is set for 5:00 PM daily, ensuring that the published report remains current without manual updates. This feature was enabled through the dataset settings after publishing the .pbix file to My Workspace.

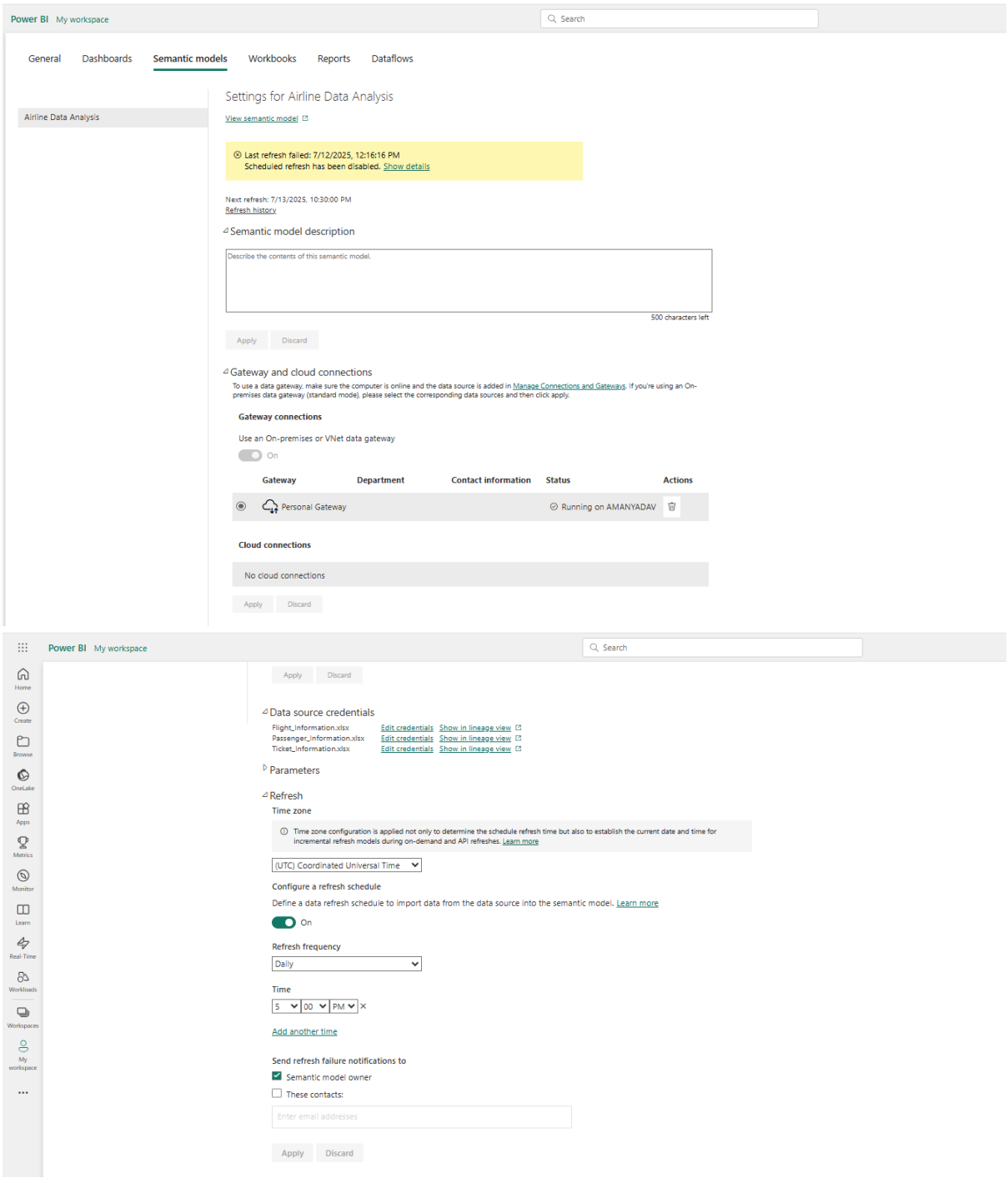


Figure 18: → Scheduled Daily Refresh Enabled at 5 PM via Default Gateway