

Airline Data Management and Analysis Using Power BI

Task 1. Data Preparation and Cleaning...

- Extract and transform data in Power Query.
- Clean data: remove duplicates, handle missing values, and format columns.

The screenshot shows the Power Query Editor interface with the following details:

- File Bar:** Home, Transform, Add Column, View, Tools, Help.
- Toolbars:** Close & Apply, New Source, Recent Data, Enter Data, Data Sources, Parameters, Refresh Preview, Advanced Editor, Properties, Manage Columns, Sort, Transform, Combine, AI Insights.
- Query Settings:** Name: flight_information, Applied Steps: Filtered Rows.
- Table:** flight_information (5 columns, 200 rows). The columns are FlightID, FlightNumber, Airline, Destination, and Status. The data includes various flight details like Houston, Chicago, New York, and different airline codes (A, B, C, D).
- Bottom:** Preview downloaded at 18:31, system tray showing 19°C Haze, ENG IN, 19:25, 23-01-2025.

The screenshot shows the Power Query Editor interface with the following details:

- File Bar:** Home, Transform, Add Column, View, Tools, Help.
- Toolbars:** Close & Apply, New Source, Recent Data, Enter Data, Data Sources, Parameters, Refresh Preview, Advanced Editor, Properties, Manage Columns, Sort, Transform, Combine, AI Insights.
- Query Settings:** Name: ticket_information, Applied Steps: Removed Duplicates.
- Table:** ticket_information (3 columns, 42 rows). The columns are FlightID, TicketID, and BookingStatus. The data includes various ticket statuses like Pending, Confirmed, and Cancelled.
- Bottom:** Preview downloaded at 18:31, system tray showing 19°C Haze, ENG IN, 19:27, 23-01-2025.

The screenshot shows the Power Query Editor interface with three queries listed: flight_information, ticket_information, and passenger_information. The passenger_information query is currently selected and displayed in the main pane as a table with columns PassengerID, FlightID, and SeatNumber. The table has 28 rows of data. In the 'APPLIED STEPS' pane, the 'Removed Duplicates' step is highlighted. The status bar at the bottom right indicates 'PREVIEW DOWNLOADED AT 18:31'.

I began by importing the three datasets: Flight Information, Passenger Information, and Ticket Information into Power BI.

I used the Power Query Editor to load and transform the data. Each table was carefully reviewed to ensure correct formats for each column.

Step 2: Clean the data

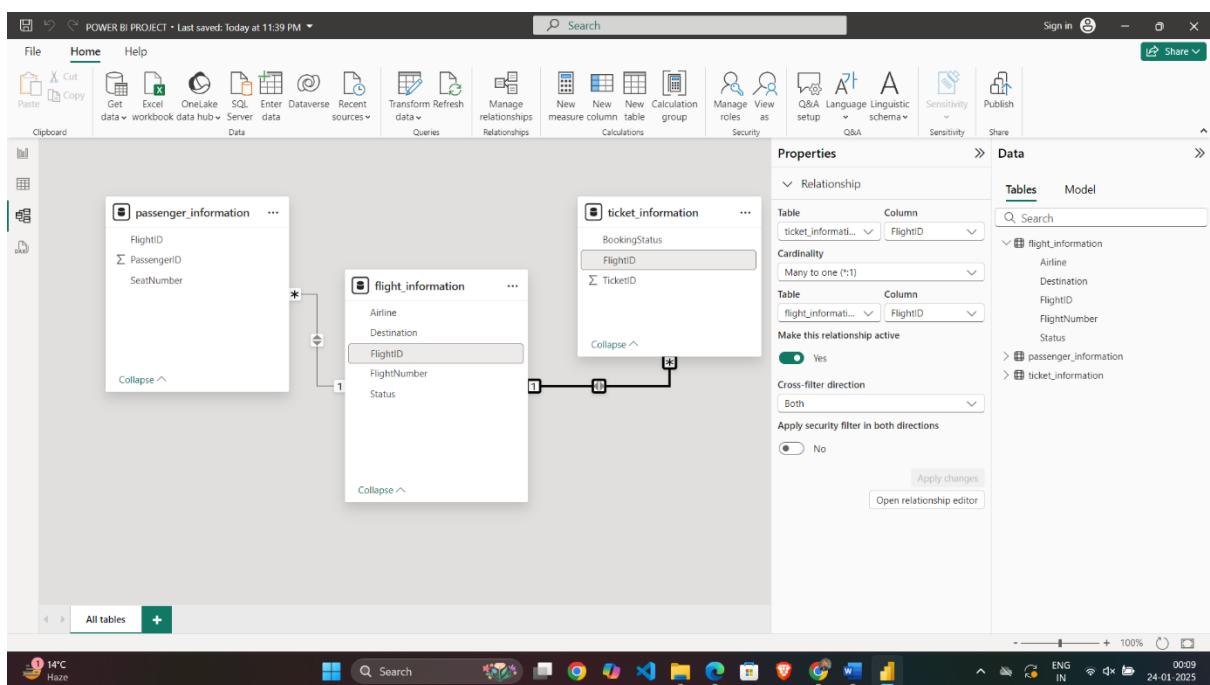
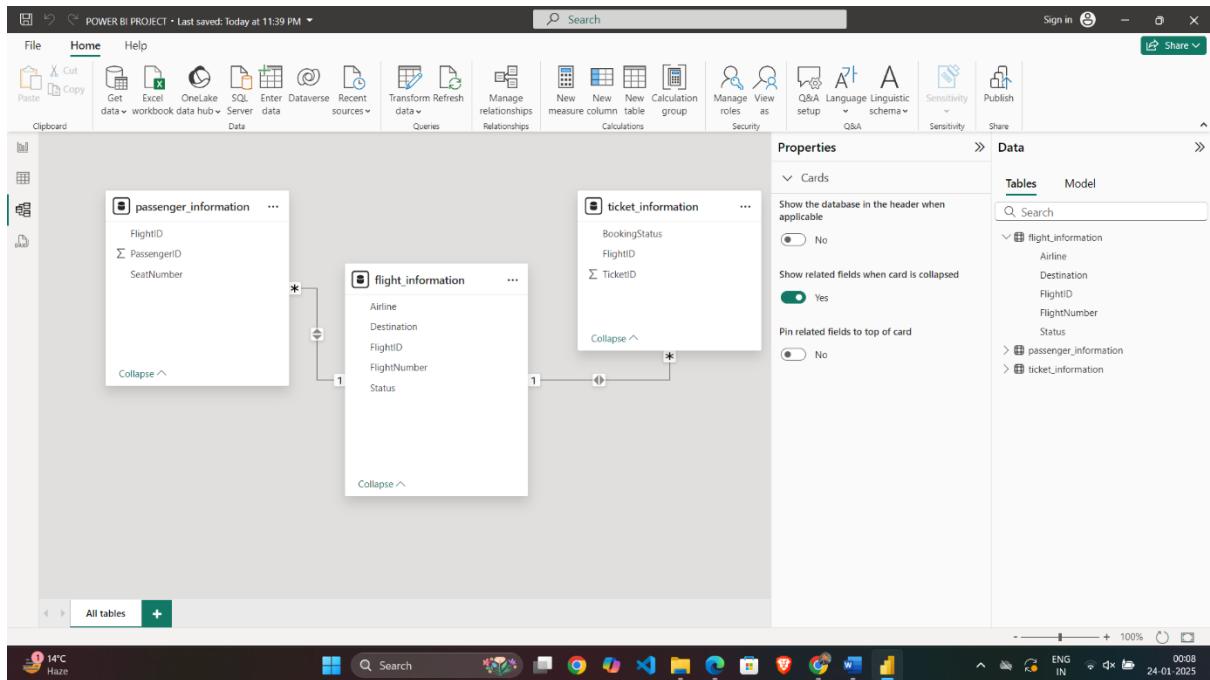
- I removed any duplicate records from each table using the "Remove Duplicates" feature in Power Query.
- For missing values, I performed the necessary actions, either filling them with appropriate default values or removing the rows where needed, ensuring data integrity.
- I formatted the columns to ensure consistency, such as converting columns to the correct data type (e.g., Date/Time, Integer).

Step 3: Deliverables

- I took a screenshot of the Power Query Editor showing the cleaned data, highlighting the changes made in the query steps.

2. Data Modeling...

Create relationships between datasets (FlightID as the key). ● Understand cardinality and configure the model appropriately.



Step 1: Create relationships between datasets

- I created relationships between the datasets using the "FlightID" column as the key, linking the Passenger Information table to the Flight Information table, and the Ticket Information table to the Flight Information table.

Step 2: Understand cardinality and configure the model

- I analyzed the cardinality of each relationship to ensure that Power BI set up the correct relationship types (e.g., One-to-Many, Many-to-One).

- I also made sure the relationships were correctly set to ensure data integrity.

Step 3: Deliverables

- I captured a screenshot of the data model, showing the relationships between the tables, with "FlightID" as the common key.

3. Enhanced Data Insights...

Add a conditional column to classify flights as "Best" or "To Be Improved" based on status.

The screenshot shows the Power BI Desktop interface with the 'Query Editor' open. The main area displays a table with columns: FlightNumber, Airline, Destination, Status, FLIGHT REMARKS, and FLIGHT NO. The data consists of 28 rows of flight information. The 'Query Settings' pane on the right shows the following applied steps:

- Source
- Navigation
- Promoted Headers
- Changed Type
- Removed Columns
- Removed Duplicates
- Changed Type1
- Filtered Rows
- Added Conditional Column
- Renamed Columns
- Inserted Text After Delimiter
- Renamed Columns1
- Filtered Rows1

At the bottom, the taskbar shows various application icons, and the system tray indicates it's 21:28 on 24-01-2025.

I created a new conditional column to classify flights as either "Best" or "To Be Improved" based on the flight's status. The status field was used to determine this classification.

- Use "Column from Examples" to extract the flight number from FlightNumber.

The screenshot shows the Power BI Data Editor interface. The main area displays a table with columns: FlightNumber, Airline, Destination, Status, Flight Remarks, and Flight No. The table contains 200 rows of flight information. The 'Query Settings' pane on the right shows the properties of the 'flight_information' table, including its name and applied steps like 'Renamed Columns1'. The status bar at the bottom indicates 'PREVIEW DOWNLOADED AT 21:27'.

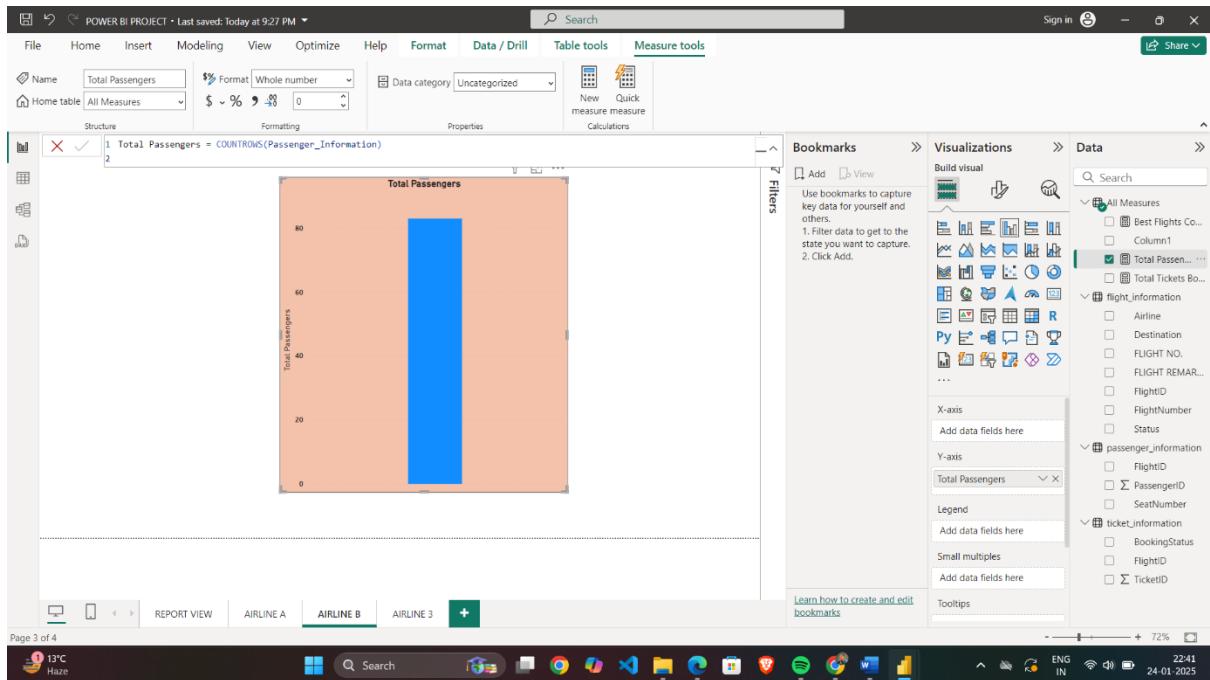
Use "Column from Examples" to extract the flight number

I used the "Column from Examples" feature to extract the flight number from the "FlightNumber" column. This helped create a clean, easily identifiable flight number.

4. Calculations Using DAX

● Calculate:

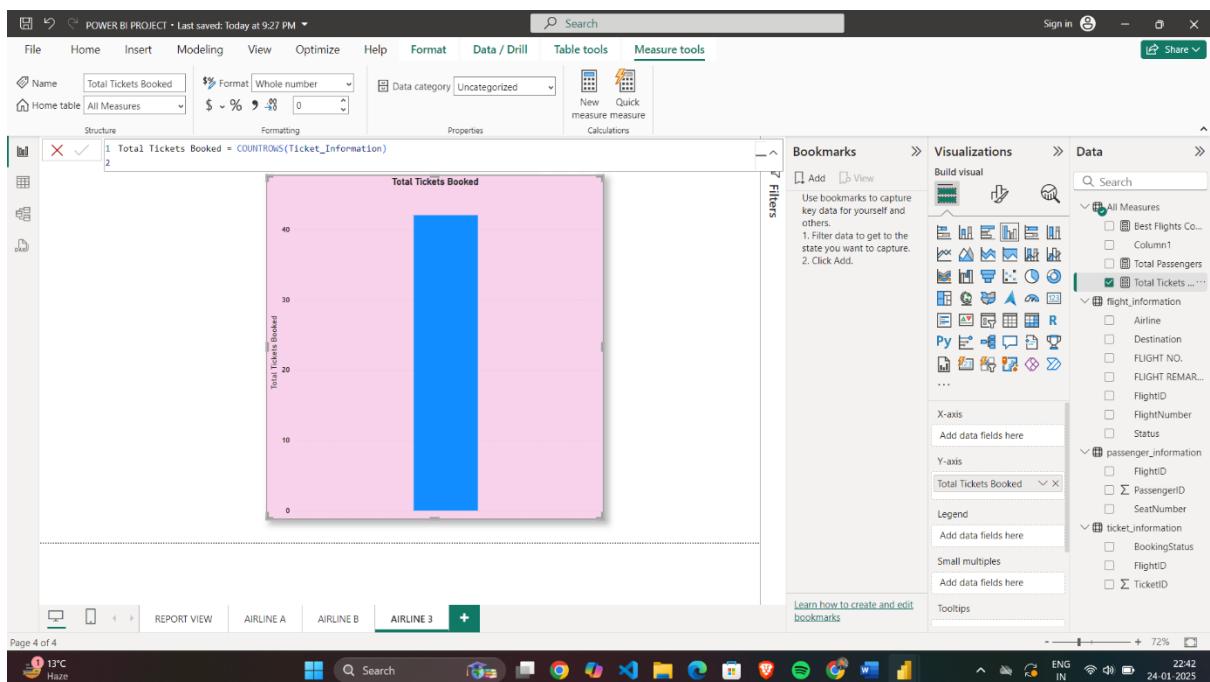
- Total passengers for a specific flight.



Calculate total passengers for a specific flight

- I created a DAX measure to calculate the total number of passengers for each specific flight by counting the number of distinct PassengerIDs for each FlightID.

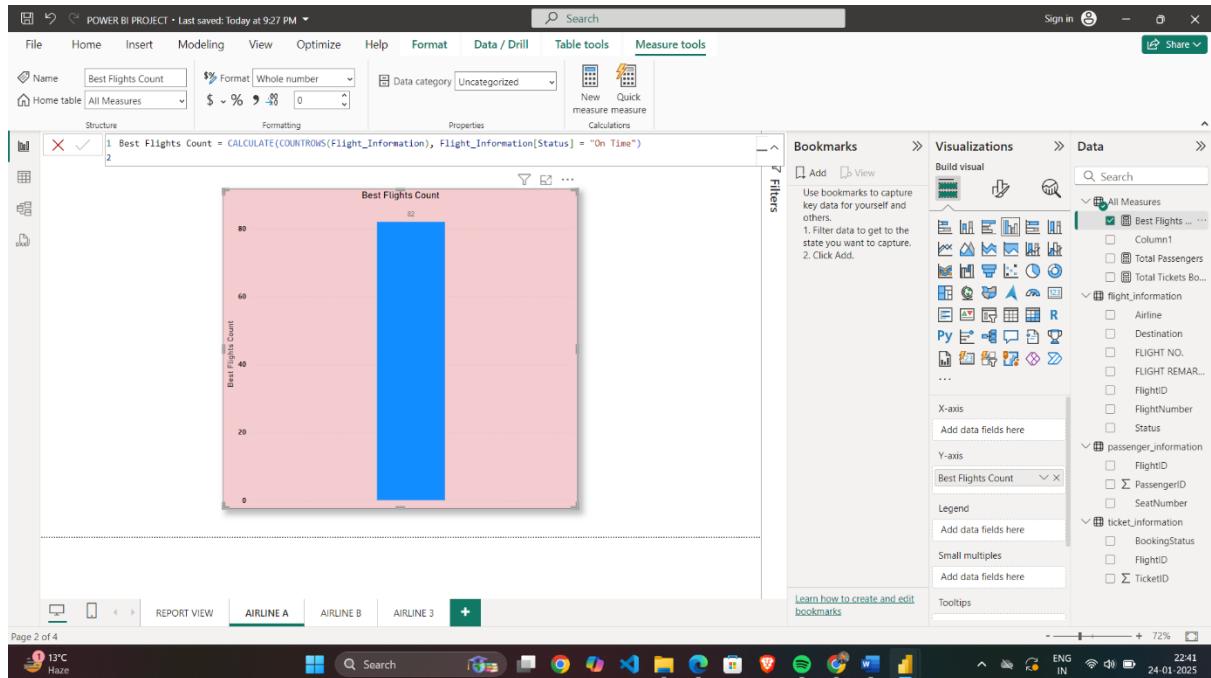
○ Total tickets booked.



Calculate total tickets booked

I created another DAX measure to calculate the total number of tickets booked for each flight by counting the TicketIDs for each FlightID in the Ticket Information table.

○ Filtered table showing "Best" flights only.

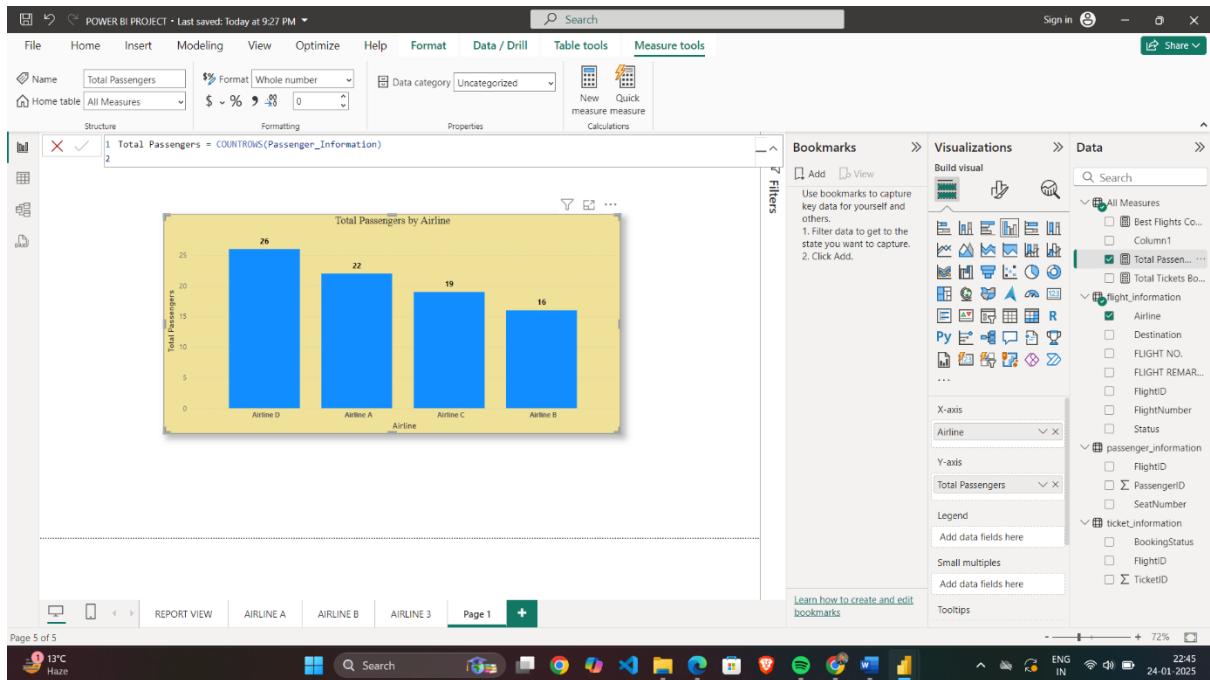


Filtered table showing "Best" flights only

I used DAX to create a calculated table that only displayed "Best" flights by filtering based on the new conditional column that classified flights.

5. Visualization and Interactive Features.

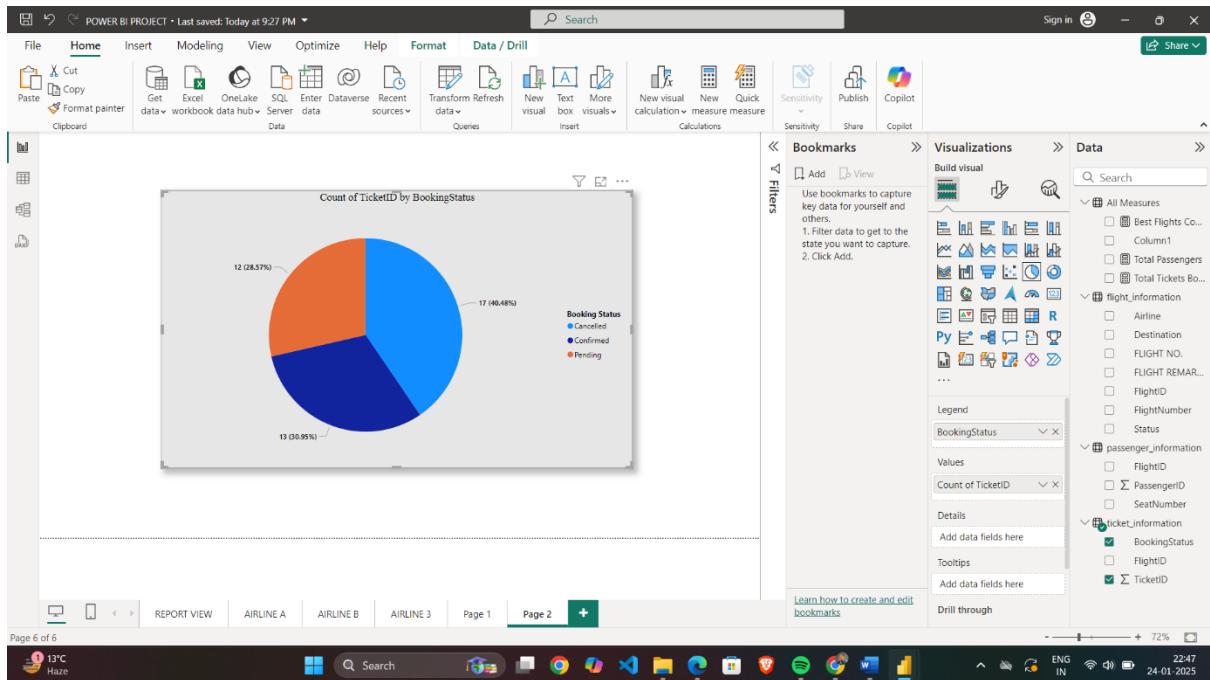
● Create visuals for: ○ Passenger count by airline.



Passenger Count by Airline (Clustered Column Chart)

- I created a Clustered Column Chart to visualize the total number of passengers by each airline.
- I used the "Passenger Information" table to count the number of passengers (using the PassengerID field).
- The Airline field was used for the Axis and the PassengerID (count) for the Values field, creating a clustered column chart where each airline has a corresponding bar showing the total number of passengers.

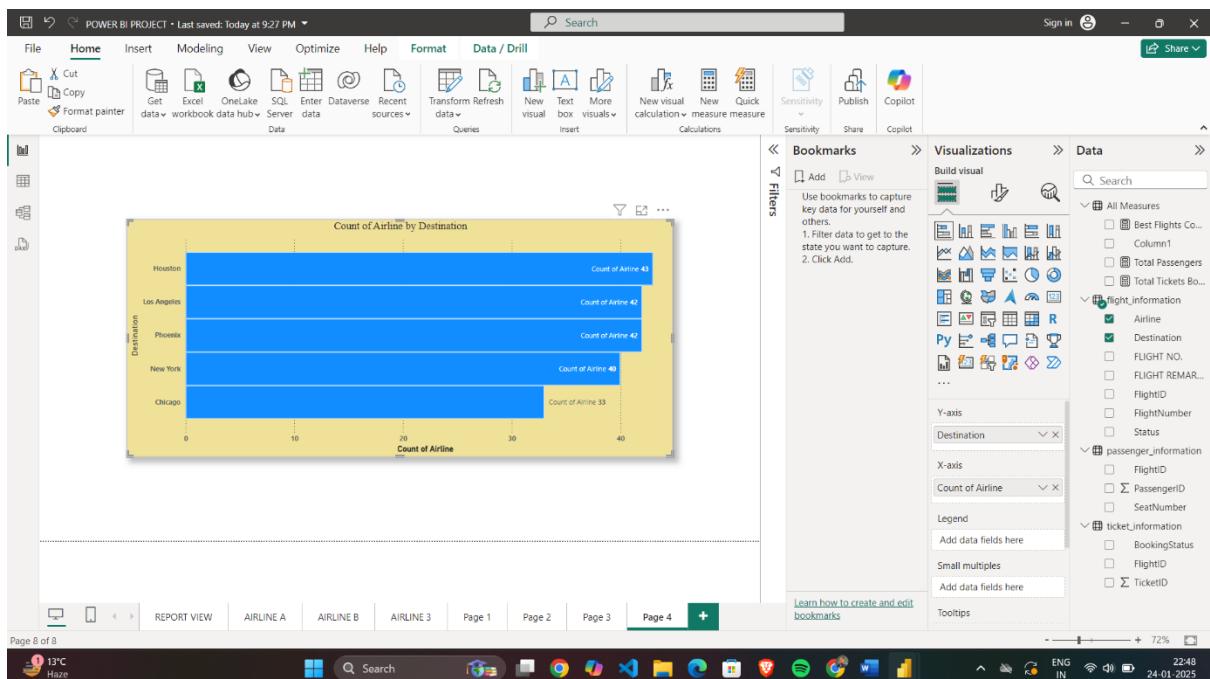
○ Ticket booking statuses.



Ticket Booking Status (Pie Chart)

- I created a Pie Chart to visualize the distribution of different ticket booking statuses, such as "Booked," "Cancelled," and "Pending."
- I used the "Ticket Information" table and categorized the data by the BookingStatus field to show the proportion of each status type in the chart.

Flights by airline and destination

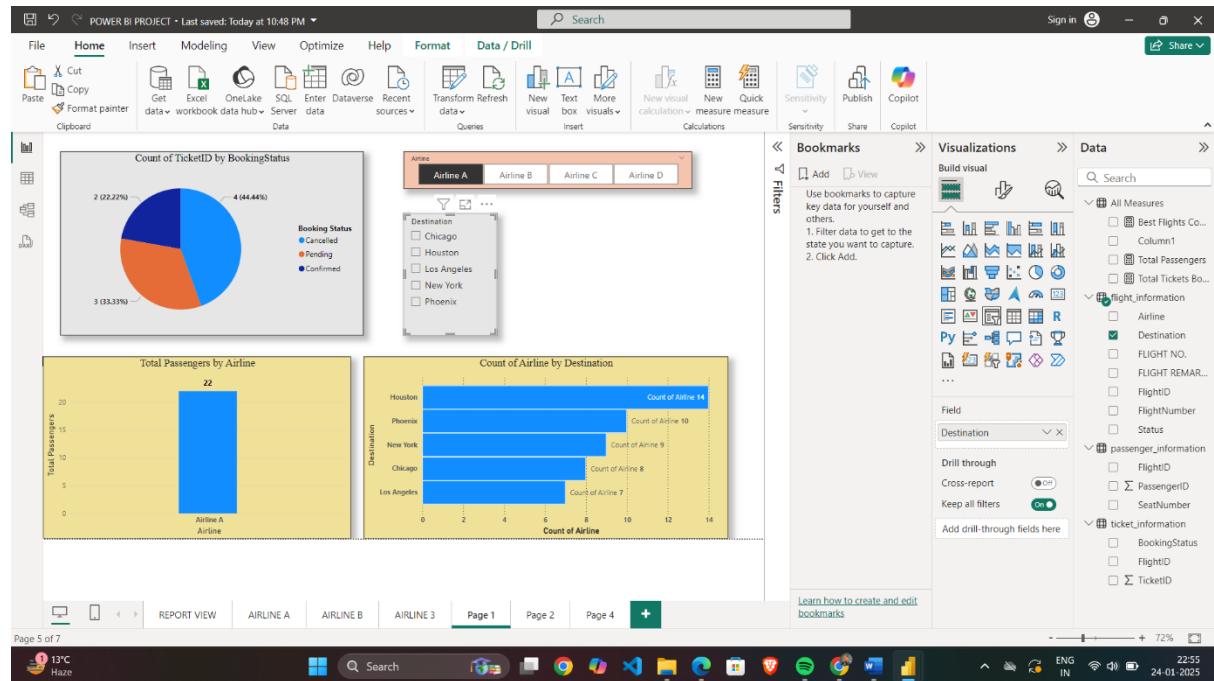


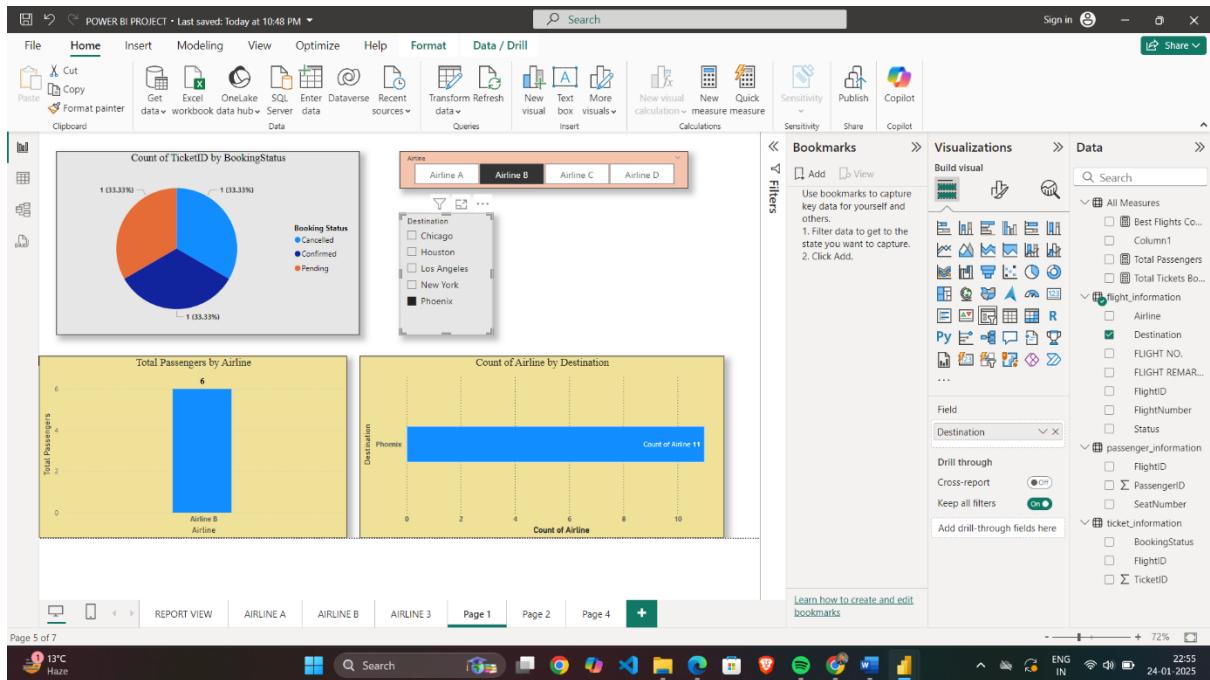
Flights by Airline and Destination (Stacked Bar Chart)

- I created a Stacked Bar Chart to visualize the flights by airline and their respective destinations.
- The Flight Information table was used, with the Airline field placed on the Axis and the Destination field in the Legend.
- The Stacked Bar Chart allows easy comparison of the number of flights by each airline across different destinations, with each bar showing how flights are distributed across multiple destinations for each airline.

● Add interactive features for:

○ Destination and Airline

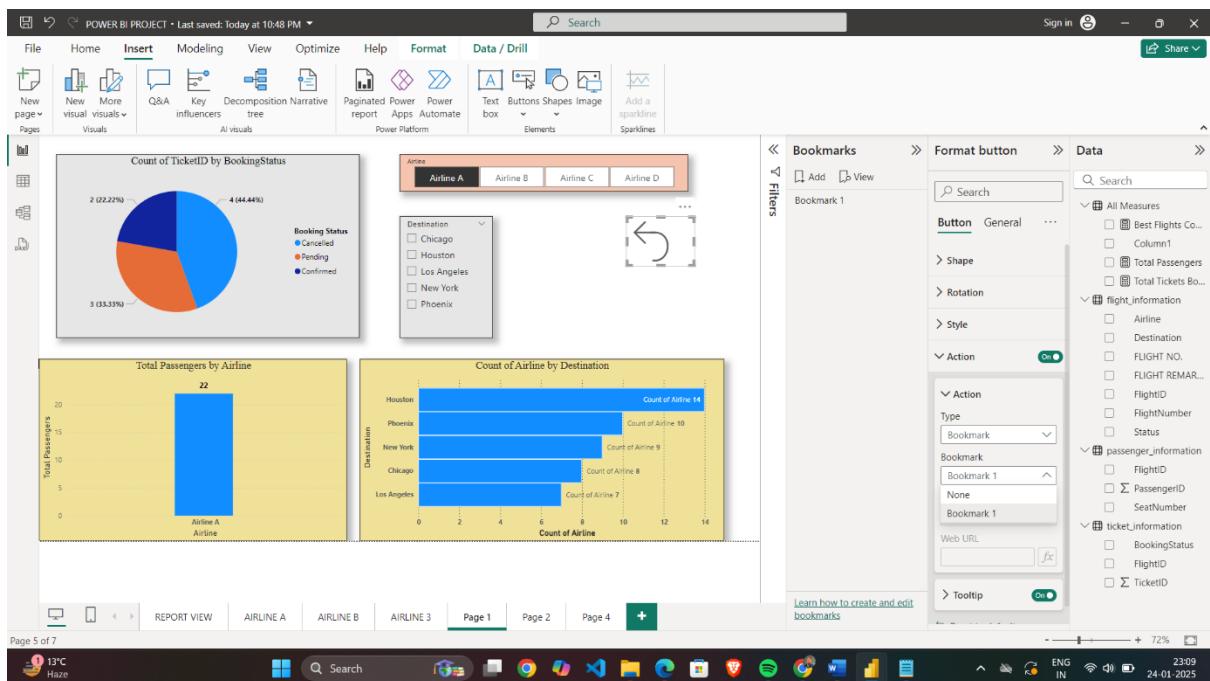


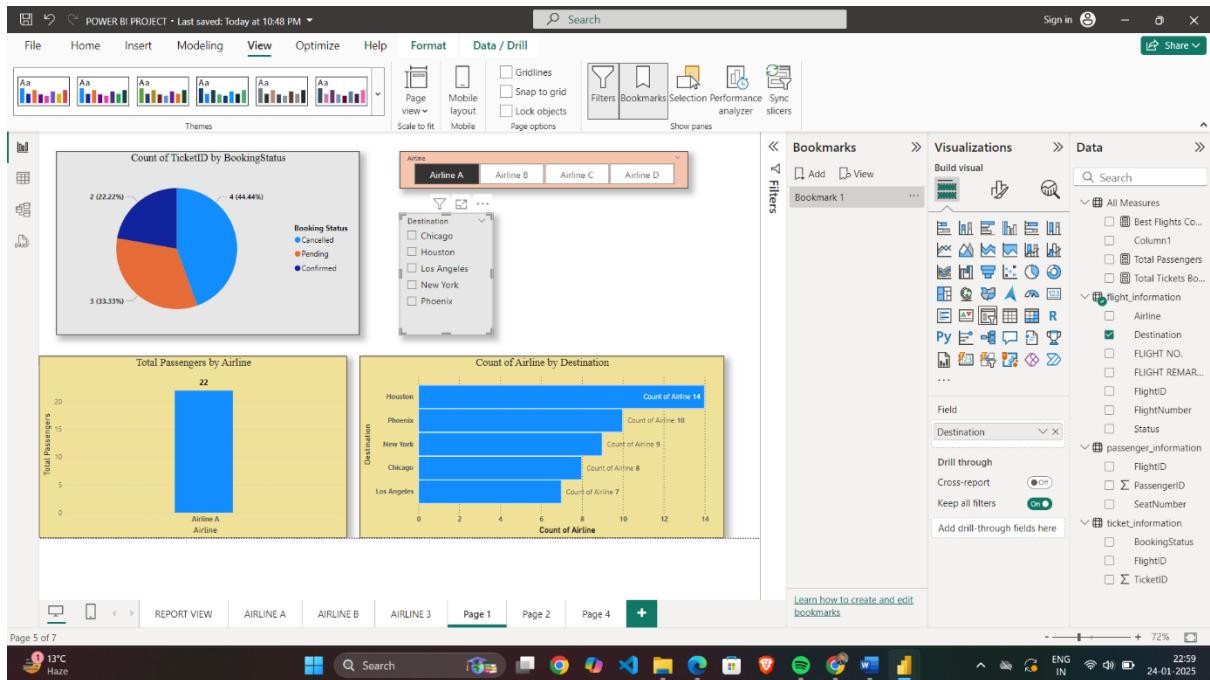


Interactive Filters and Slicers

- I added interactive slicers for the Destination and Airline fields.
- This allows users to filter the data dynamically, enabling them to view specific destinations or airlines and see how the data in the visuals changes based on their selections.

Quickviews.

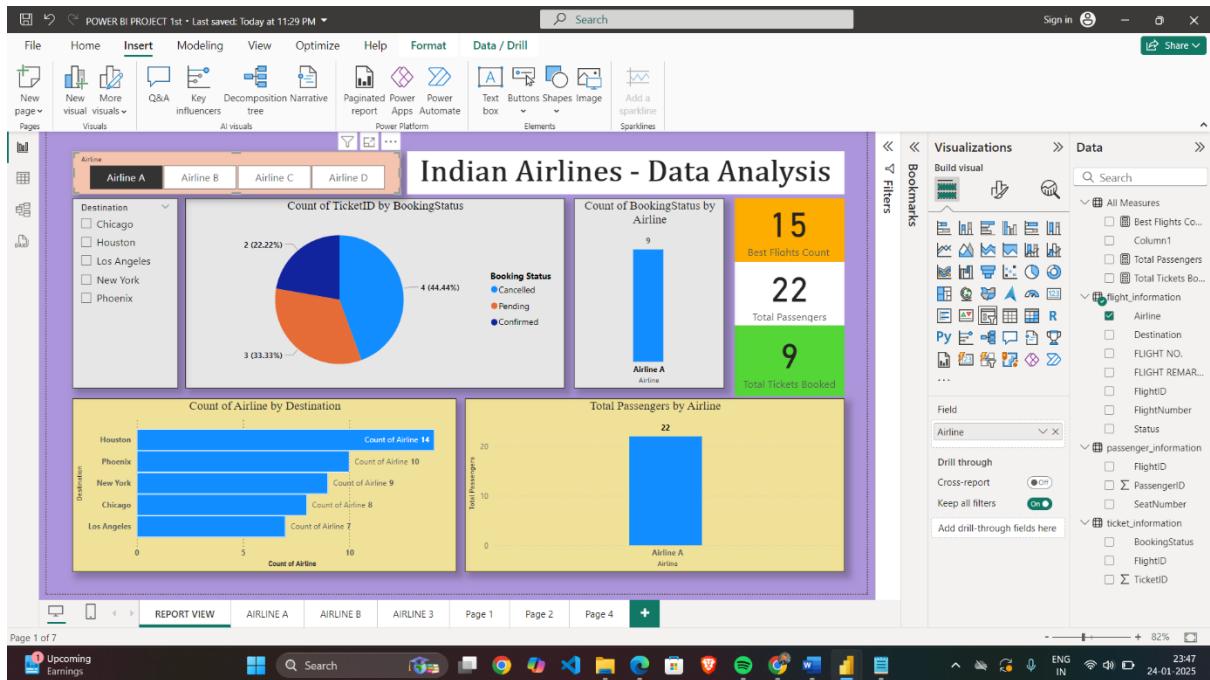




Quick Views for Easy Access to Insights

- I used buttons to create Quick Views for users, allowing them to quickly toggle between key insights on the dashboard.
- For example, I created buttons to quickly view:
 - Passenger Count by Airline
 - Ticket Booking Status
 - Flights by Airline and Destination
- These buttons allow users to seamlessly navigate between different sections of the dashboard, providing easy access to important information without the need to scroll or search for specific insights.
- Each button was assigned to a specific bookmark, which stores the state of the visuals, allowing for quick switching between different views of the data.

○ Airline-specific pages.

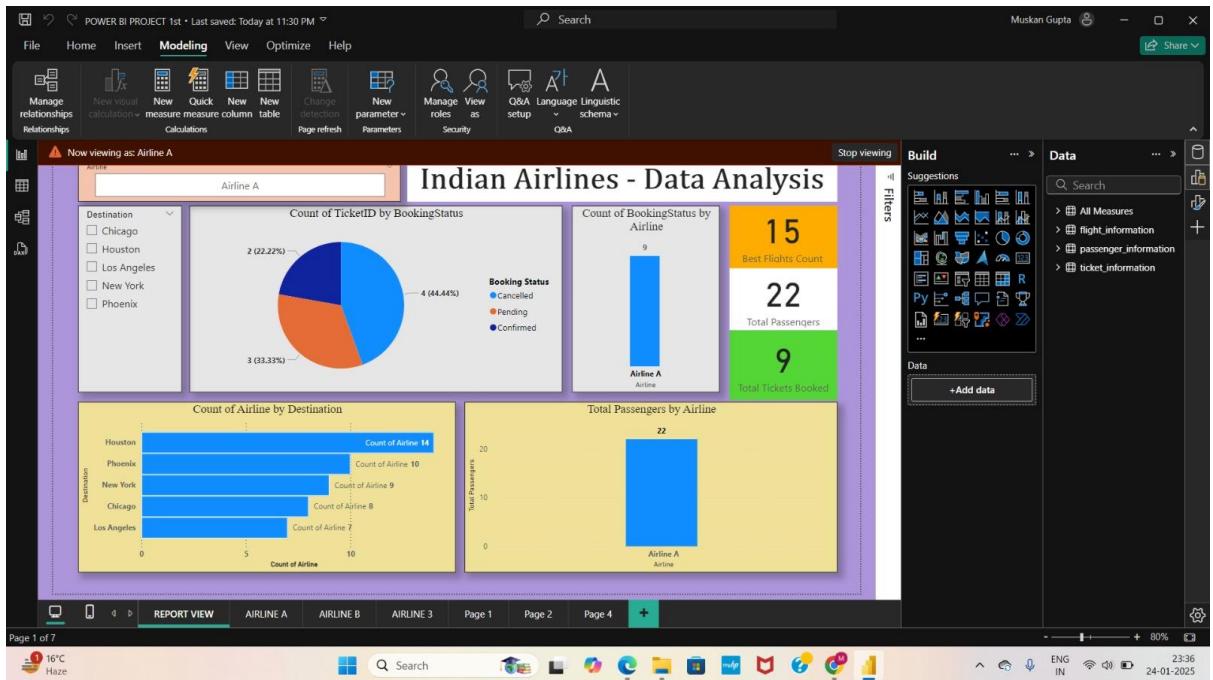


Airline-Specific Pages

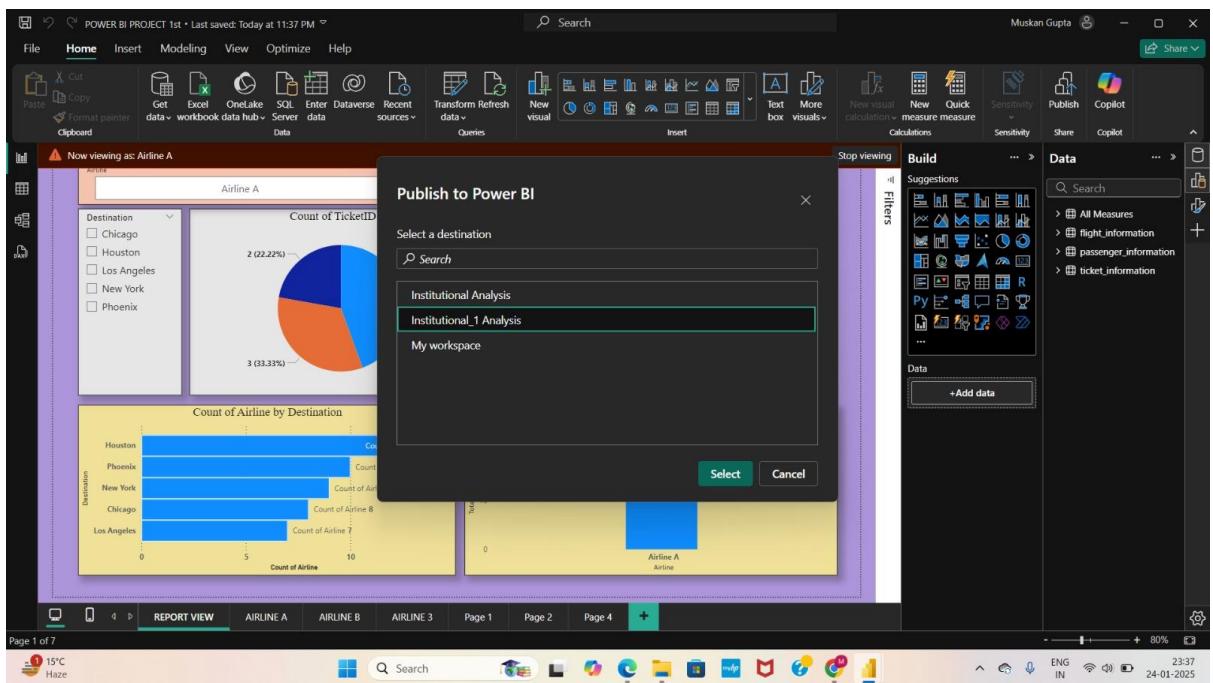
- I created airline-specific pages that allow users to drill down into data specific to each airline.
- By selecting a particular airline, users are directed to a detailed page with focused analysis, including total passengers, booking statuses, and flight destinations

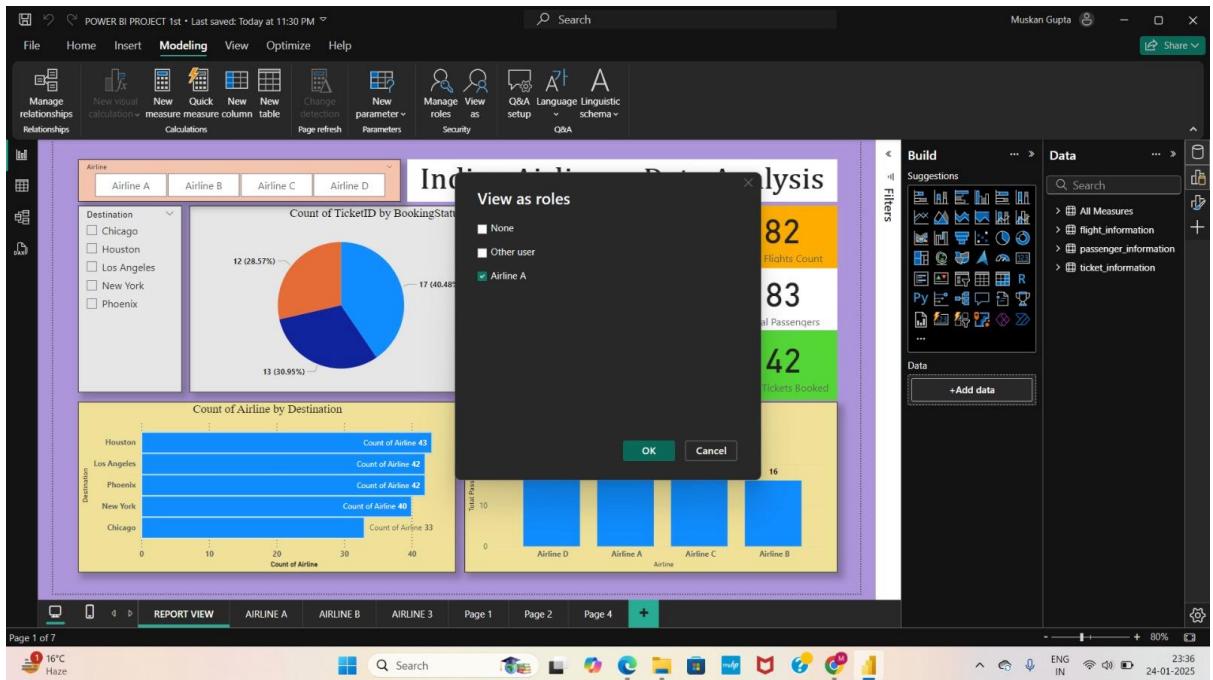
6. Final Dashboard and Power BI Service

- Design a comprehensive dashboard with key visuals and insights.

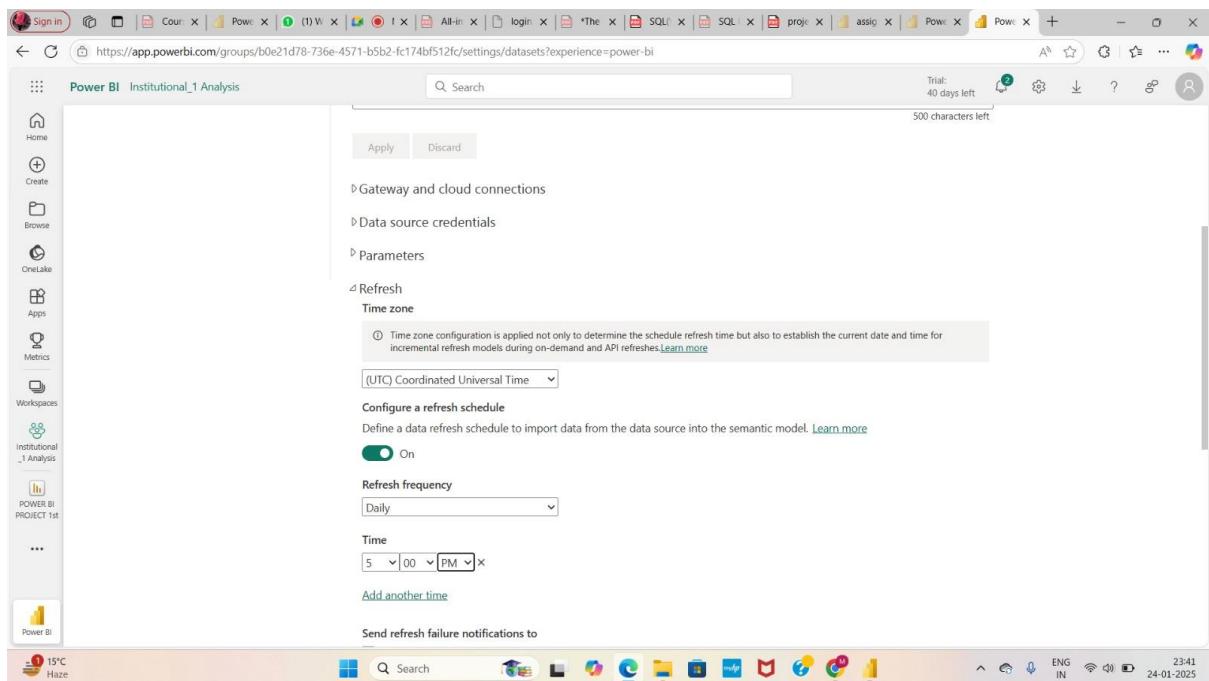


● Configure Row-Level Security (RLS) for Airline A data and assign it to a user.





● Setup a schedule refresh at 5 PM daily



Step 1: Design a comprehensive dashboard

- I designed a dashboard that included all the key visuals and insights such as passenger count, booking statuses, and flights by airline and destination.
- The dashboard was structured to provide clear, actionable insights for stakeholders.

Step 2: Configure Row-Level Security (RLS)

- I set up Row-Level Security (RLS) to ensure that users from Airline A could only see data specific to their airline.
- I created security roles and applied filters to restrict access to data based on the user's role (Airline A in this case).

Step 3: Set up scheduled refresh

- I configured a scheduled refresh to update the data daily at 5 PM to ensure the dashboard always reflects the most recent data.