

Airline Data Management and Analysis

TO ANALYZE AND VISUALIZE AIRLINE DATA FOR OPERATIONAL INSIGHTS, PASSENGER MANAGEMENT, AND TICKET BOOKING TRENDS USING POWER BI.

BY
YUGMA PATEL

1. Data Preparation and Cleaning

Tasks Performed:

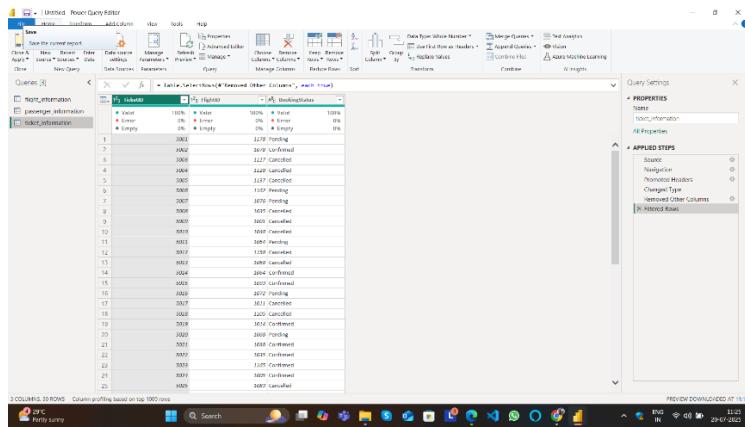
- Data was extracted and transformed using Power Query Editor.
- Cleaned the data by:
 - Removing duplicate rows.
 - Handling missing values.
 - Formatting columns like FlightID, PassengerID, and TicketID.

FlightInformation

| FlightID | Origin | Destination | Airline |
|--------------|-----------|-------------|-----------|
| 30501 FL1020 | Atlanta D | Seattle | Airline A |
| 30501 FL1020 | Atlanta D | Seattle | Airline A |
| 30501 FL1080 | Airline A | New York | Canceled |
| 30501 FL1070 | Airline C | Chicago | Delayed |
| 30501 FL1070 | Airline C | New York | Unscaled |
| 30501 FL1070 | Airline C | Phoenix | On Time |
| 30501 FL1070 | Airline C | Los Angeles | Canceled |
| 30501 FL1000 | Airline C | Los Angeles | Delayed |
| 30501 FL1040 | Airline C | Chicago | On Time |
| 30501 FL1040 | Airline A | Phoenix | Canceled |
| 30501 FL1040 | Airline A | New York | On Time |
| 30501 FL1040 | Airline C | New York | Delayed |
| 30501 FL1040 | Airline C | Phoenix | Unscaled |
| 30501 FL1040 | Airline C | Los Angeles | Unscaled |
| 30501 FL1040 | Airline C | Seattle | Delayed |
| 30501 FL1040 | Airline B | New York | Delayed |
| 30501 FL1040 | Airline B | Seattle | Delayed |
| 30501 FL1070 | Airline B | Seattle | Delayed |
| 30501 FL1070 | Airline B | Chicago | Canceled |
| 30501 FL1070 | Airline B | New York | On Time |
| 30501 FL1070 | Airline B | Seattle | Delayed |
| 30501 FL1070 | Airline C | New York | Unscaled |
| 30501 FL1070 | Airline C | Seattle | On Time |
| 30501 FL1070 | Airline D | Phoenix | On Time |

Passenger

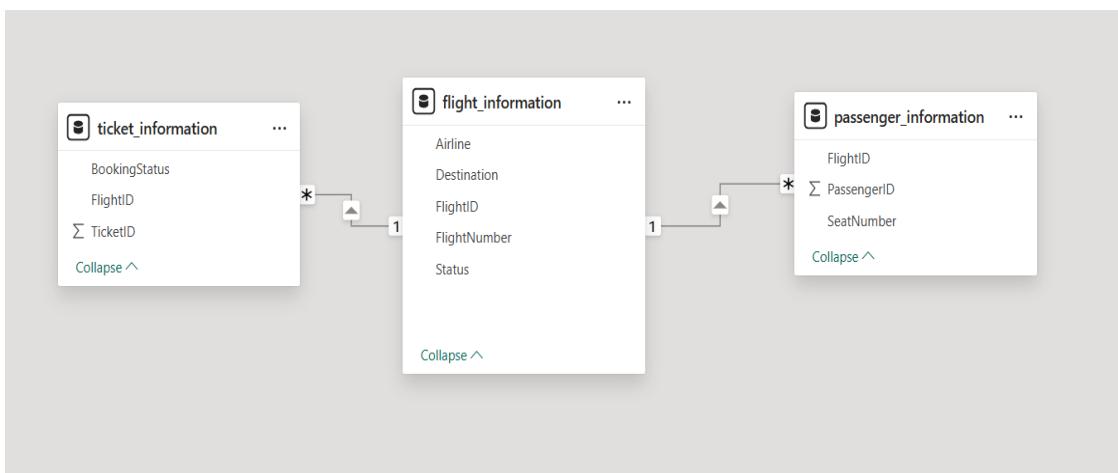
| PassengerID | BirthDate |
|-------------|------------|
| 1 | 1921-01-01 |
| 2 | 1924-08-01 |
| 3 | 1906-01-01 |
| 4 | 1929-02-01 |
| 5 | 1929-02-01 |
| 6 | 1929-02-01 |
| 7 | 1902-01-01 |
| 8 | 1915-07-01 |
| 9 | 1927-04-01 |
| 10 | 1927-04-01 |
| 11 | 1923-04-01 |
| 12 | 1923-04-01 |
| 13 | 1920-02-01 |
| 14 | 1920-02-01 |
| 15 | 1900-01-01 |
| 16 | 1995-03-01 |
| 17 | 1995-03-01 |
| 18 | 1995-03-01 |
| 19 | 1922-01-01 |
| 20 | 1917-03-01 |
| 21 | 1917-03-01 |
| 22 | 1940-08-01 |
| 23 | 1927-08-01 |
| 24 | 1927-08-01 |
| 25 | 1927-08-01 |



2. DATA MODELING

Tasks Performed:

- Created data model by establishing relationships:
 - FlightID** is the primary key linking:
 - flight_information
 - passenger_information
 - ticket_information
- Configured cardinality and relationship direction correctly.



3. ENHANCED DATA INSIGHT

Tasks Performed:

- Created **Conditional Column** to classify flights as:
 - "Best" or "To Be Improved" based on Status.

The screenshot shows the 'Add Conditional Column' dialog in Power BI. The 'New column name' is 'Flight Rating'. The logic is defined as follows:

- If Status equals 'On Time' Then 'Best'
- Else If Status equals 'Delayed' Then 'To be improved'
- Else 'To be improved'

The underlying query is: `Table.SelectColumns(#"Changed Type",["FlightID", "FlightNumber", "Airline", "Destination", "Status"])`

| FlightID | FlightNumber | Airline | Destination | Status |
|----------|--------------|-----------|-------------|-----------|
| 1029 | FL1683 | Airline B | Chicago | Cancelled |
| 1020 | FL1130 | Airline A | New York | On Time |
| 1021 | FL1661 | Airline B | New York | Cancelled |
| 1022 | FL1308 | Airline A | Houston | Delayed |
| 1023 | FL1769 | Airline A | Chicago | On Time |

- Used **Column From Examples** to extract values like:
 - Flight Number from FlightNumber.

The screenshot shows the 'Add Column From Examples' dialog. The 'Flight No' column is being extracted from the 'FlightNumber' column. The dialog displays sample rows and the resulting column structure.

| FlightID | FlightNumber | Airline | Destination | Status | Flight No |
|----------|--------------|-----------|-------------|-----------|-----------------------|
| 1 | 1001 | Airline D | Houston | On Time | 1102 |
| 2 | 1002 | Airline B | Chicago | On Time | FL1102 (FlightNumber) |
| 3 | 1003 | Airline A | New York | Cancelled | |
| 4 | 1004 | Airline C | Chicago | Delayed | |
| 5 | 1005 | Airline B | New York | Delayed | |
| 6 | 1006 | Airline A | Phoenix | On Time | |
| 7 | 1007 | Airline C | Los Angeles | Cancelled | |
| 8 | 1008 | Airline C | Los Angeles | Delayed | |
| 9 | 1009 | Airline A | Los Angeles | Cancelled | |
| 10 | 1010 | Airline D | Chicago | Cancelled | |
| 11 | 1011 | Airline A | Phoenix | On Time | |
| 12 | 1012 | Airline D | New York | Delayed | |
| 13 | 1013 | Airline C | Houston | On Time | |
| 14 | 1014 | Airline C | New York | Delayed | |
| 15 | 1015 | Airline C | Houston | Delayed | |
| 16 | 1016 | Airline B | New York | Delayed | |
| 17 | 1017 | Airline D | Phoenix | Delayed | |
| 18 | 1018 | Airline B | Houston | Delayed | |
| 19 | 1019 | Airline B | Chicago | Cancelled | |
| 20 | 1020 | Airline A | New York | On Time | |
| 21 | 1021 | Airline B | New York | Cancelled | |
| 22 | 1022 | Airline A | Houston | Delayed | |

4. CALCULATIONS USING DAX

Tasks Performed:

DAX Measures Created:

- Total passengers = DISTINCTCOUNT(passenger_information[PassengerID])
- Total tickets booked = COUNTROWS(ticket_information)
- Total Flights = DISTINCTCOUNT(flight_information[FlightID])
- Best Flight
CALCULATE(COUNTROWS(flight_information),flight_information[Flight Rating] = "Best")
- DelayedCount =
CALCULATE(COUNTROWS(flight_information),flight_information[Status] = "Delayed")

```

1 Total tickets booked =
2 CALCULATE(
3     COUNTROWS(ticket_information),
4     ticket_information[BookingStatus] = "Confirmed"
5 )

```

| re | Formatting | Properties | Calculations |
|----|---|------------|--------------|
| . | Best Flight = CALCULATE(COUNTROWS(flight_information),flight_information[Flight Rating] = "Best") | | |

5. VISUALIZATION AND INTERACTIVE FEATURES

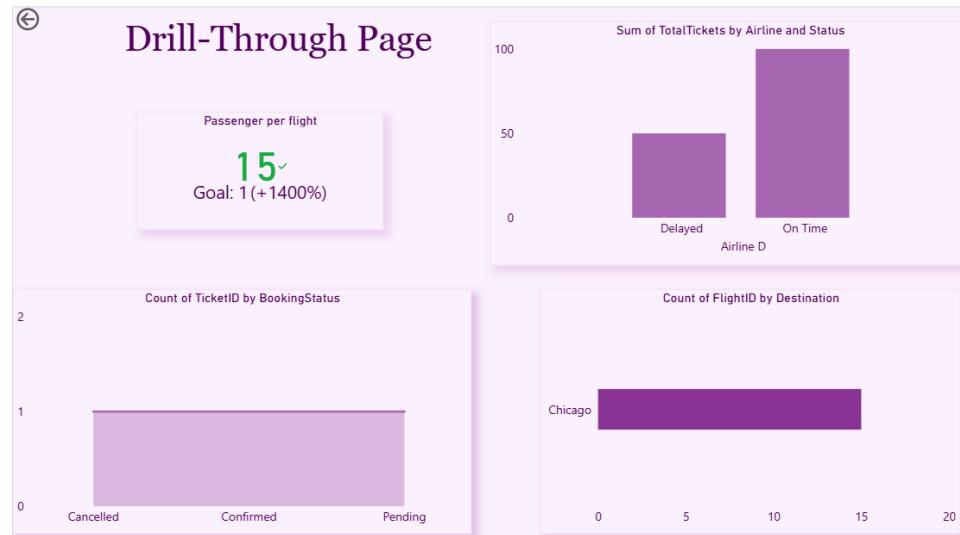
Visuals Created:

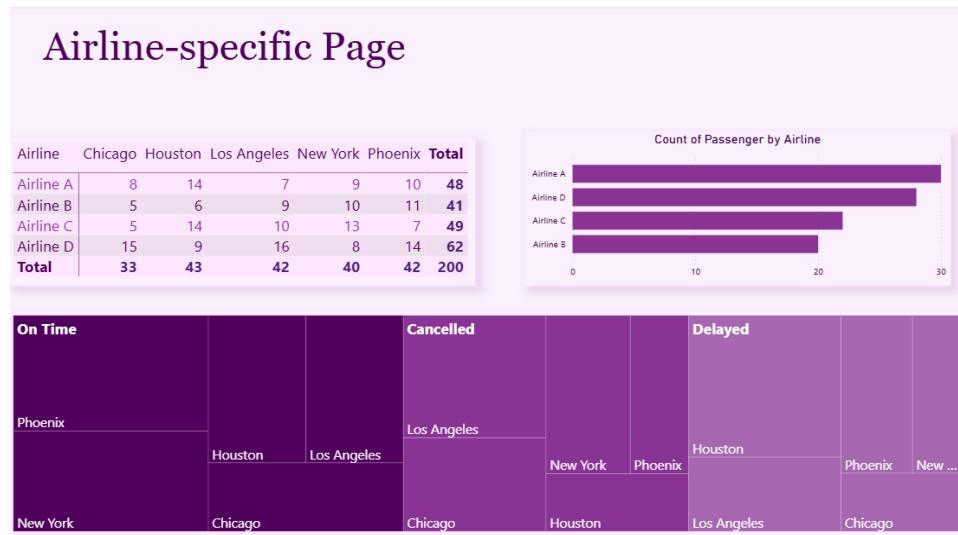
- Passenger count by airline (Donut Chart).
- Ticket booking statuses (Pie Chart).
- Flights by airline and destination (Stacked Bar).
- Card: Most Popular Destination, Total Flights, Tickets Booked, Total Passengers.
- Drill-through Page:
 - Breakdown by Airline and Destination
- Summary Page:
 - Table: Distribution of Best Airlines per destination.
 - Donut: Passenger count by airline.

Interactive Features:

- Slicers for:
 - Airline
 - Destination
- Drill-through for Airline-specific details.

- Reset filter button using bookmarks.





6. COMPREHENSIVE DASHBOARD DESIGN

Objective:

To build a visually insightful and interactive dashboard that presents key metrics, trends, and breakdowns for better airline management and decision-making.

Key Visuals and Features Implemented:

1. Cards (Top Metrics):

- **Total Flights** – Displays the total number of flights analyzed (e.g., 200).
- **Total Passengers** – Total count of passengers across all flights (e.g., 100).
- **Total Tickets Booked** – Sum of ticket records in the data (e.g., 17).
- **Most Popular Destination** – Highlights the city with the highest flight count (e.g., Chicago).

Purpose: To give management a quick overview of essential performance indicators.

2. Slicers (Interactive Filters):

- Airline slicer (Airline A, B, C, D).
- Destination slicer (Chicago, Houston, etc.).

- **Clear All Slicers** button added for better usability.

Purpose: Allows users to filter visuals dynamically based on selections, aiding in focused analysis.

3. Charts and Visuals:

- **Donut Chart:** Passenger count by airline (shows % distribution clearly).
- **Pie Chart:** Ticket booking status – Confirmed, Cancelled, Pending.
- **Stacked Bar Chart:** Flights categorized by Airline and Destination.
- **Line/Area Chart:** Visualizing “Best” flights per airline based on OnTime status.
- **Gauge/Bar:** Visual to show OnTime performance.

Purpose: These visuals together offer performance, trends, and comparisons.

4. Drill-through Page:

- Provides a deep dive into a selected airline or destination.
- Shows metrics like passenger per flight, count by booking status, flight count by destination.

Purpose: Enables detailed analysis when a specific airline or flight is selected.

5. Summary Page:

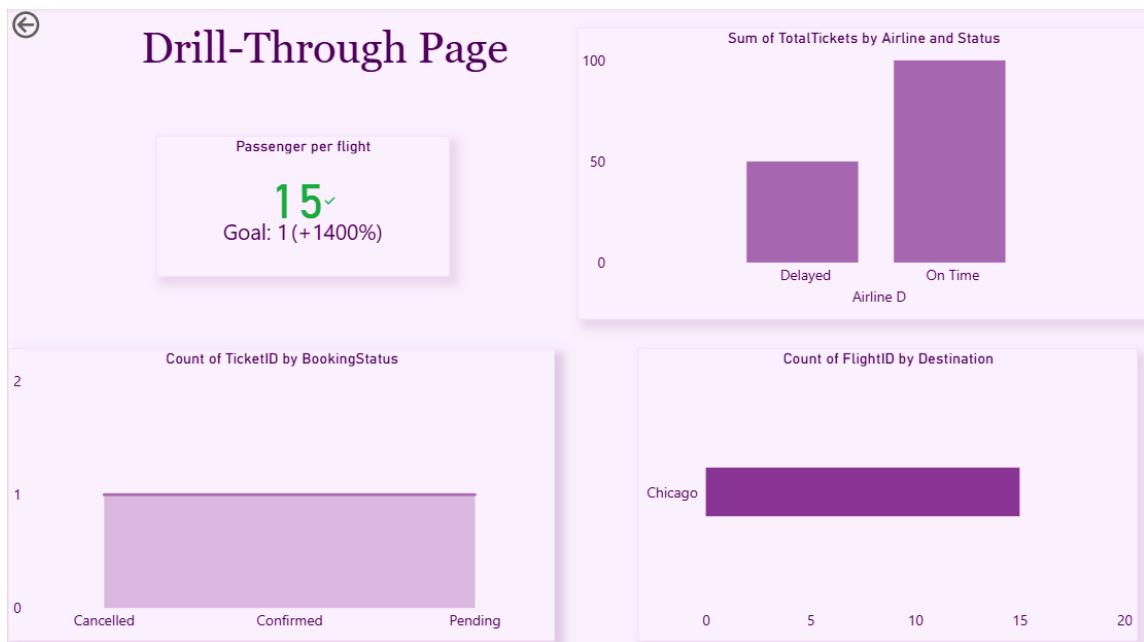
- **Donut chart** showing passenger distribution by airline.
- **Matrix table** for Best Flights distribution across cities.

Purpose: Offers a consolidated view of all airline operations across cities.

Airline Data Management & Analysis



Drill-Through Page



Airline-specific Page

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



7. CONFIGURE ROW-LEVEL SECURITY (RLS) FOR AIRLINE A DATA AND ASSIGN IT TO A USER.

Row-Level Security (RLS) Implementation

Objective:

To restrict data access such that users only see data relevant to **Airline A**.

Steps Performed:

1. Created RLS Role:

- In Power BI Desktop, went to the **Modeling** tab and selected **Manage Roles**.
- Created a new role called "AirlineA_Only".

2. Defined the DAX Filter:

- Applied this filter on the flight_information table:

DAX

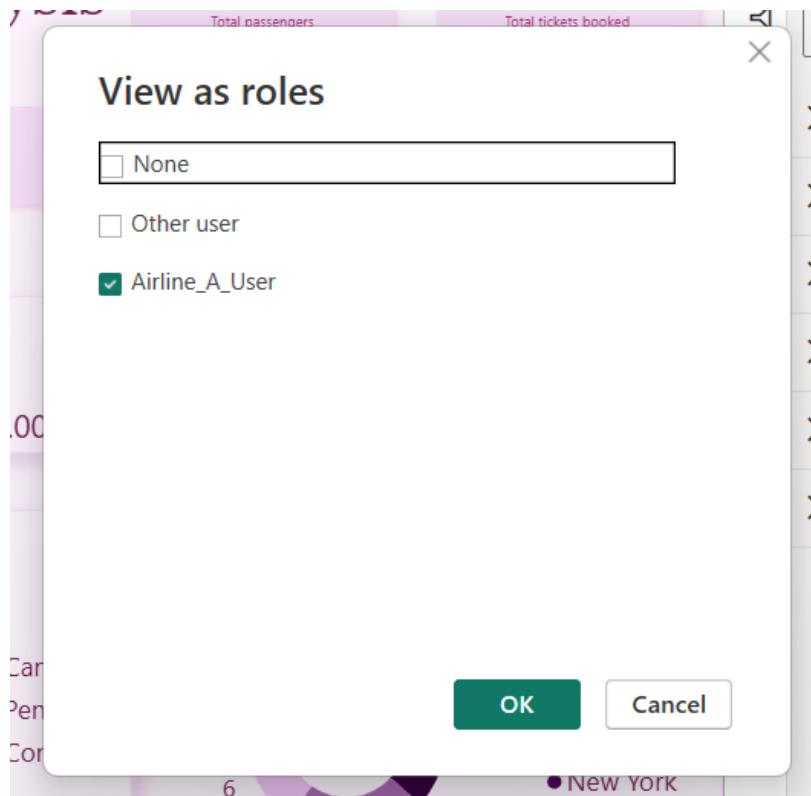
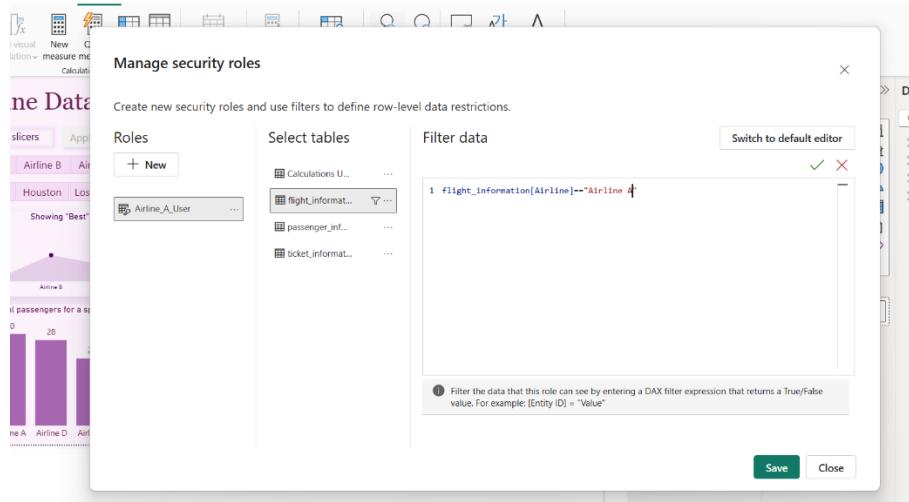
```
flight_information[Airline]=="Airline A"
```

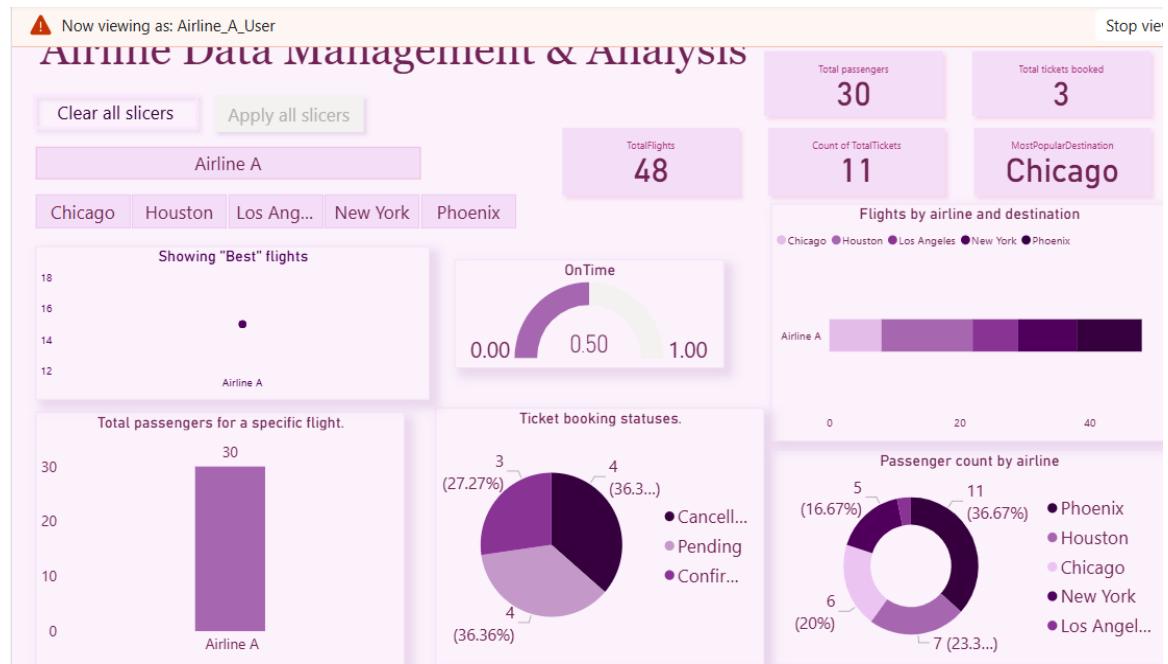
3. Tested the Role:

- Used the **View As Role** option to verify that only Airline A's data is visible.

4. Published to Power BI Service:

- After publishing, assigned this role to a specific user under **Dataset Security** in Power BI Service.





app.powerbi.com/groups/6fb71bee-9061-4d2d-9d19-89bc10e034e1/rowlevelsecurity/1146776?experience=power-bi

Swayam Central Gmail YouTube Maps Adobe Acrobat

Power BI Personal work space > Row-Level Security

Row-Level Security

Airline_A_User (1)

Members (1)

People or groups who belong to this role

Enter email addresses

Add

YUGMA PATEL

Save Cancel

8. SETUP AS SCHEDULE REFRESH AT 5 PM DAILY.

Objective:

To automate data updates every day at 5 PM IST.

Steps Performed:

- Published Report to Power BI Service.

2. Went to the **Workspace > Dataset > Settings > Scheduled Refresh**.
3. Enabled the scheduled refresh and selected the time:
 Daily at **5:00 PM (IST)**.
4. Ensured **credentials were configured** correctly for the data source connection.

