

Airline Data Management and Analysis using Power Bi

2025

Submitted By:
Saraswathi
Gongala

Problem Statement:

The airline industry operates with numerous complexities, requiring effective data management and insights into flight schedules, Passenger details, and ticketing systems. This project aims to analyze airline operations for improving efficiency and customer Satisfaction.

Task 1: Data Preparation and Cleaning

During this process, I loaded Power Query Editor in Power BI with the three datasets (Flight_Information, Passenger_Information, and Ticket_Information).

Steps Taken:

- Imported data into Power Query for each table.
- Removed duplicates from all datasets (i.e. duplicate passenger or flight records).
- Dealt with missing values: For example, null Status fields are replaced with "Unknown", null or blank Seat-Number entries were filled with NA
- Formatted the column data types (i.e. FlightDate columns converted to Date, PassengerID converted to text).
- Renamed columns for readability and consistency (i.e. FlightID to Flight ID, BookingStatus to Booking Status).

Flight ID	Flight Name	Flight Number	Destination	Airline Type	Status	Performance Status
1001	FL1102	1102	Houston	D	On Time	Best
1002	FL1435	1435	Chicago	B	On Time	Best
1003	FL1860	1860	New York	A	Cancelled	Poor
1004	FL1270	1270	Chicago	C	Delayed	To Be Improved
1005	FL1106	1106	New York	C	Delayed	To Be Improved
1006	FL1071	1071	Phoenix	A	On Time	Best
1007	FL1700	1700	Los Angeles	C	Cancelled	Poor
1008	FL1020	1020	Los Angeles	C	Delayed	To Be Improved
1009	FL1614	1614	Los Angeles	A	Cancelled	Poor
1010	FL1121	1121	Chicago	D	Cancelled	Poor
1011	FL1466	1466	Phoenix	A	On Time	Best
1012	FL1214	1214	New York	D	Delayed	To Be Improved
1013	FL1330	1330	Houston	C	On Time	Best
1014	FL1458	1458	New York	C	Delayed	To Be Improved
1015	FL1087	1087	Houston	C	Delayed	To Be Improved
1016	FL1372	1372	New York	B	Delayed	To Be Improved
1017	FL1099	1099	Phoenix	D	Delayed	To Be Improved
1018	FL1871	1871	Houston	B	Delayed	To Be Improved
1019	FL1663	1663	Chicago	B	Cancelled	Poor
1020	FL1130	1130	New York	A	On Time	Best
1021	FL1661	1661	New York	B	Cancelled	Poor
1022	FL1308	1308	Houston	A	Delayed	To Be Improved
1023	FL1769	1769	Chicago	A	On Time	Best
1024	FL1343	1343	Chicago	B	Delayed	To Be Improved
1025	FL1491	1491	Phoenix	D	On Time	Best
1026	FL1413	1413	Chicago	D	Cancelled	Poor
1027	FL1805	1805	Chicago	D	On Time	Best
1028	FL1385	1385	Chicago	D	On Time	Best

Passenger ID	Flight ID	Seat Number
1	1161	38A
2	1157	24D
3	1141	30B
4	1046	17E
5	1035	29D
6	1134	10A
7	1082	10A
8	1115	20E
9	1197	34E
10	1047	2E
11	1153	43C
12	1194	48C
13	1010	47A
14	1056	23C
15	1030	16D
16	1109	40D
17	1005	25C
18	1119	32C
19	1033	27E
20	1118	32B
21	1065	19E
22	1146	5B
23	1177	28B
24	1011	22E
25	1085	6A
26	1026	5A
27	1063	12B

	Ticket ID	Flight ID	Booking Status
1	5001	1178	Pending
2	5002	1078	Confirmed
3	5003	1117	Cancelled
4	5004	1120	Cancelled
5	5005	1137	Cancelled
6	5006	1162	Pending
7	5007	1076	Pending
8	5008	1035	Cancelled
9	5009	1001	Cancelled
10	5010	1040	Cancelled
11	5011	1064	Pending
12	5012	1150	Cancelled
13	5013	1060	Cancelled
14	5014	1064	Confirmed
15	5015	1093	Confirmed
16	5016	1072	Pending
17	5017	1011	Cancelled
18	5018	1105	Cancelled
19	5019	1014	Confirmed
20	5020	1060	Pending
21	5021	1030	Confirmed
22	5022	1035	Confirmed
23	5023	1165	Confirmed
24	5024	1005	Confirmed
25	5025	1083	Cancelled
26	5026	1123	Cancelled
27	5027	1078	Confirmed

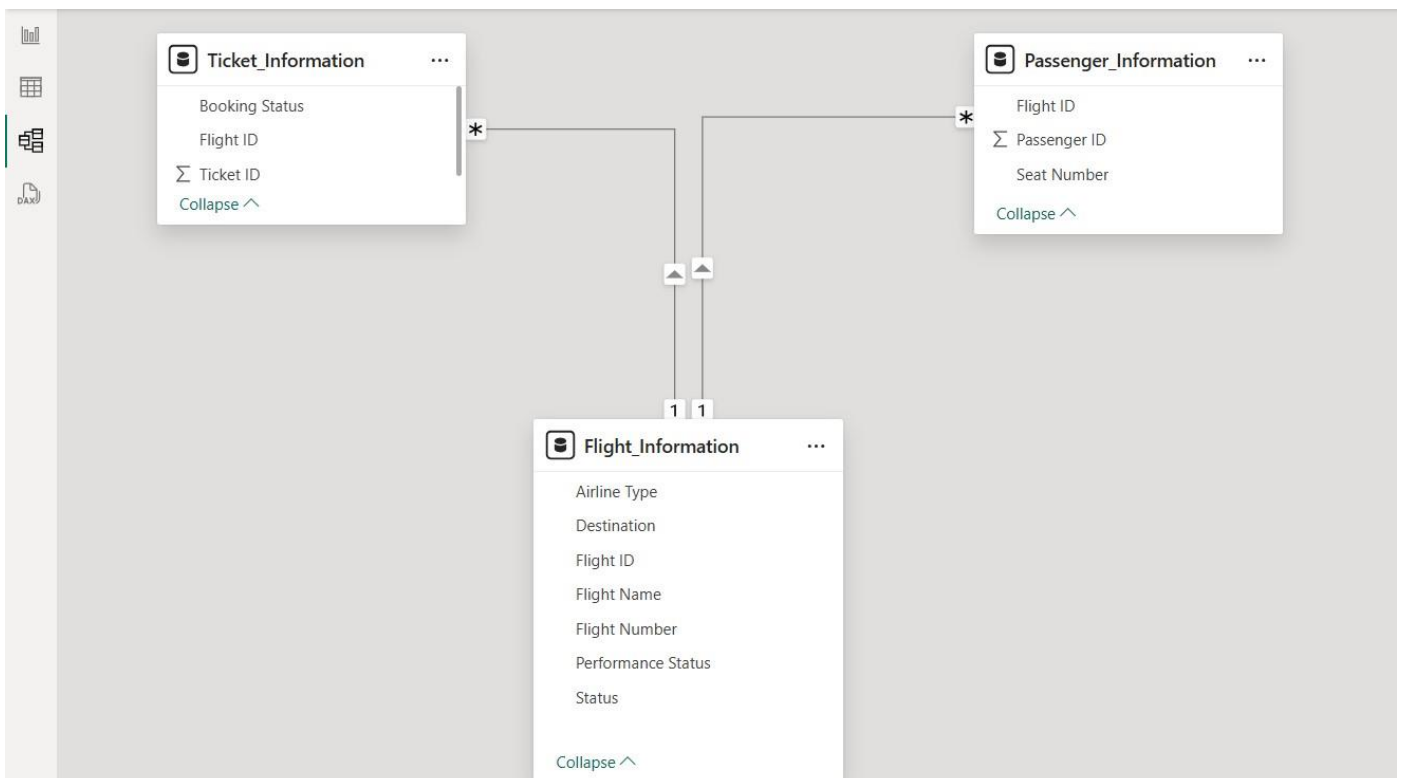
Task 2: Data Modeling

After cleaning the data, I uploaded it into Power BI. Then I created a data model, using the Flight ID field as the primary key to relate the three tables.

Steps Completed:

Connected:

- **Flight_Information[Flight ID] ↔ Passenger_Information[Flight ID]**
- **Flight_Information[Flight ID] ↔ Ticket_Information[Flight ID]**
- Used One-to-Many relationships when needed.
- Verified the cardinality and referential integrity to validate accurate filtering and aggregated results across related tables.



Task 3: New Insights - Enrichment of Data Insights

In this task I focused on data enrichment by adding new insights using various features of Power Query.

Enhancements:

- Conditional Column: Added a column called “Performance Status” which classified individual flights:

If the Status = "On Time" → Best

“Delayed” → to be improved

“Canceled” → Poor

- Column from Examples: 'Column from Examples' feature of Power BI was used to extract the flight number (e.g numeric code) from Flight Number and renamed the original as Flight Name.

Flight ID	Flight Name	Flight Number	Destination	Airline Type	Status	Performance Status
1001	FL1102	1102	Houston	D	On Time	Best
1002	FL1435	1435	Chicago	B	On Time	Best
1003	FL1860	1860	New York	A	Cancelled	Poor
1004	FL1270	1270	Chicago	C	Delayed	To Be Improved
1005	FL1106	1106	New York	C	Delayed	To Be Improved
1006	FL1071	1071	Phoenix	A	On Time	Best
1007	FL1700	1700	Los Angeles	C	Cancelled	Poor
1008	FL1020	1020	Los Angeles	C	Delayed	To Be Improved
1009	FL1614	1614	Los Angeles	A	Cancelled	Poor
1010	FL1121	1121	Chicago	D	Cancelled	Poor
1011	FL1466	1466	Phoenix	A	On Time	Best
1012	FL1214	1214	New York	D	Delayed	To Be Improved
1013	FL1330	1330	Houston	C	On Time	Best
1014	FL1458	1458	New York	C	Delayed	To Be Improved
1015	FL1087	1087	Houston	C	Delayed	To Be Improved
1016	FL1372	1372	New York	B	Delayed	To Be Improved
1017	FL1099	1099	Phoenix	D	Delayed	To Be Improved
1018	FL1871	1871	Houston	B	Delayed	To Be Improved
1019	FL1663	1663	Chicago	B	Cancelled	Poor
1020	FL1130	1130	New York	A	On Time	Best
1021	FL1661	1661	New York	B	Cancelled	Poor
1022	FL1308	1308	Houston	A	Delayed	To Be Improved
1023	FL1769	1769	Chicago	A	On Time	Best
1024	FL1343	1343	Chicago	B	Delayed	To Be Improved
1025	FL1491	1491	Phoenix	D	On Time	Best
1026	FL1413	1413	Chicago	D	Cancelled	Poor
1027	FL1805	1805	Chicago	D	On Time	Best
1028	FL1385	1385	Chicago	D	On Time	Best

Task 4: Calculations Using DAX

In this task, I was able to use DAX (Data Analysis Expressions) for creating measures and calculated tables to create certain metrics that were meaningful and dynamic for the analysis.

Key Calculations:

1-Total Passengers for a Specific Flight

I created a measure to calculate the number of passengers associated with a specific flight using the CALCULATE function. First I created a “Total Passengers” Measure to calculate the total passengers. After that used calculate function and filtered flight ID to get a specific result, making it interactive and context-aware.

DAX: Total Passengers for specific flight = CALCULATE ([Total passengers], Passenger_Information[Flight ID])

2-Total Tickets Booked

I used to the “DISTINCTCOUNT” function to determine the total number of unique tickets in the data. This will prevent counting any duplicate “Ticket ID” which keeps the booking totals accurate.

DAX: Total Tickets = DISTINCTCOUNT(Ticket_Information[Ticket ID])

Flight ID	Total Passengers for specific flight	Total ticket
1014		2
1040		2
1042		1
1060		2
1062		1
1064		2
1076		1
1089		2
1093		1
1097		1
1104		1
1117		1
1120		1
1125		1
1132		1
1137		1
1143		1
1150		1
1178		1
1001	1	1
1004	1	
1005	1	1
1006	1	
1010	1	
1011	1	1
1012	1	
1023	1	1
1026	1	
1027	1	
1032	1	1
1033	1	
1034	1	
1035	1	2
1038	1	
Total	100	50

3. Filtered Table: Best Flights

To do an analysis of only the best-performing flights, I created a new DAX table (not a visual filter or a measure). This table filters the “Flight Information” dataset to include only the records in which Performance Status equals "Best".

DAX: Best Flights = FILTER(Flight_Information,Flight_Information[Performance Status] = "Best")

Flight ID	Flight Name	Flight Number	Destination	Airline Type	Status	Performance Status
1001	FL1102	1102	Houston	D	On Time	Best
1002	FL1435	1435	Chicago	B	On Time	Best
1006	FL1071	1071	Phoenix	A	On Time	Best
1011	FL1466	1466	Phoenix	A	On Time	Best
1013	FL1330	1330	Houston	C	On Time	Best
1020	FL1130	1130	New York	A	On Time	Best
1023	FL1769	1769	Chicago	A	On Time	Best
1025	FL1491	1491	Phoenix	D	On Time	Best
1027	FL1805	1805	Chicago	D	On Time	Best
1028	FL1385	1385	Chicago	D	On Time	Best
1029	FL1191	1191	Los Angeles	D	On Time	Best
1030	FL1955	1955	Phoenix	B	On Time	Best
1031	FL1276	1276	New York	B	On Time	Best
1033	FL1459	1459	New York	D	On Time	Best
1034	FL1313	1313	Phoenix	B	On Time	Best
1036	FL1252	1252	Phoenix	D	On Time	Best
1039	FL1560	1560	Chicago	B	On Time	Best
1043	FL1681	1681	Houston	C	On Time	Best
1044	FL1475	1475	Phoenix	B	On Time	Best
1046	FL1975	1975	Chicago	D	On Time	Best
1048	FL1189	1189	New York	A	On Time	Best
1050	FL1686	1686	Phoenix	C	On Time	Best
1052	FL1562	1562	Phoenix	D	On Time	Best
1053	FL1875	1875	Chicago	C	On Time	Best
1055	FL1243	1243	New York	B	On Time	Best
1057	FL1504	1504	Phoenix	A	On Time	Best
1060	FL1818	1818	Chicago	D	On Time	Best
1061	FL1646	1646	Los Angeles	D	On Time	Best

Task 5: Visualizations and Interactive Options

This task was centered on converting the cleaned and modeled data into interactive visualizations or dashboards that let users slice data in different views while providing insights into the performance of airlines, passenger demographic spread, and booking patterns.

◆ Visuals Designed:

Passenger Count by Airline – This was communicated with a Clustered column chart showing the total number of passengers that each airline served.

Ticket Booking Statuses – This was illustrated using a donut chart showing the totals of tickets that were booked, cancelled, and pending.

Flights by Airline and Destination – This was visualized using a clustered column chart which shows the number of flights each airline has operated to each destination.



◆ Interactivity Options Developed:

Slicers were built to allow users to filter data by:

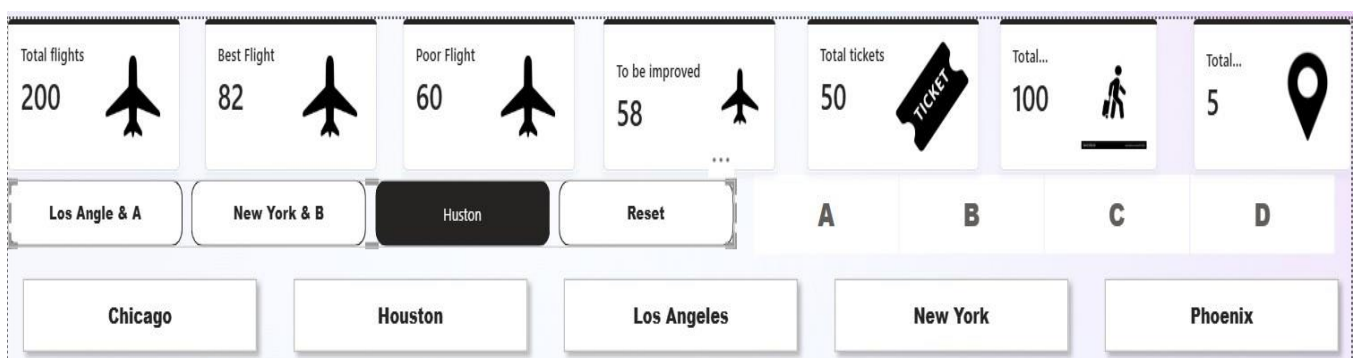
- Destination
- Airline

Bookmarks were built to create a simulated Quick Views with bookmarks such as:

- Location & Airline type

Drillthrough Pages were also created to drill down into each airline's dashboard.

- For example, if "Airline A" was clicked, they would navigate to the page where that airlines performance metrics, passengers, and bookings will be shown.



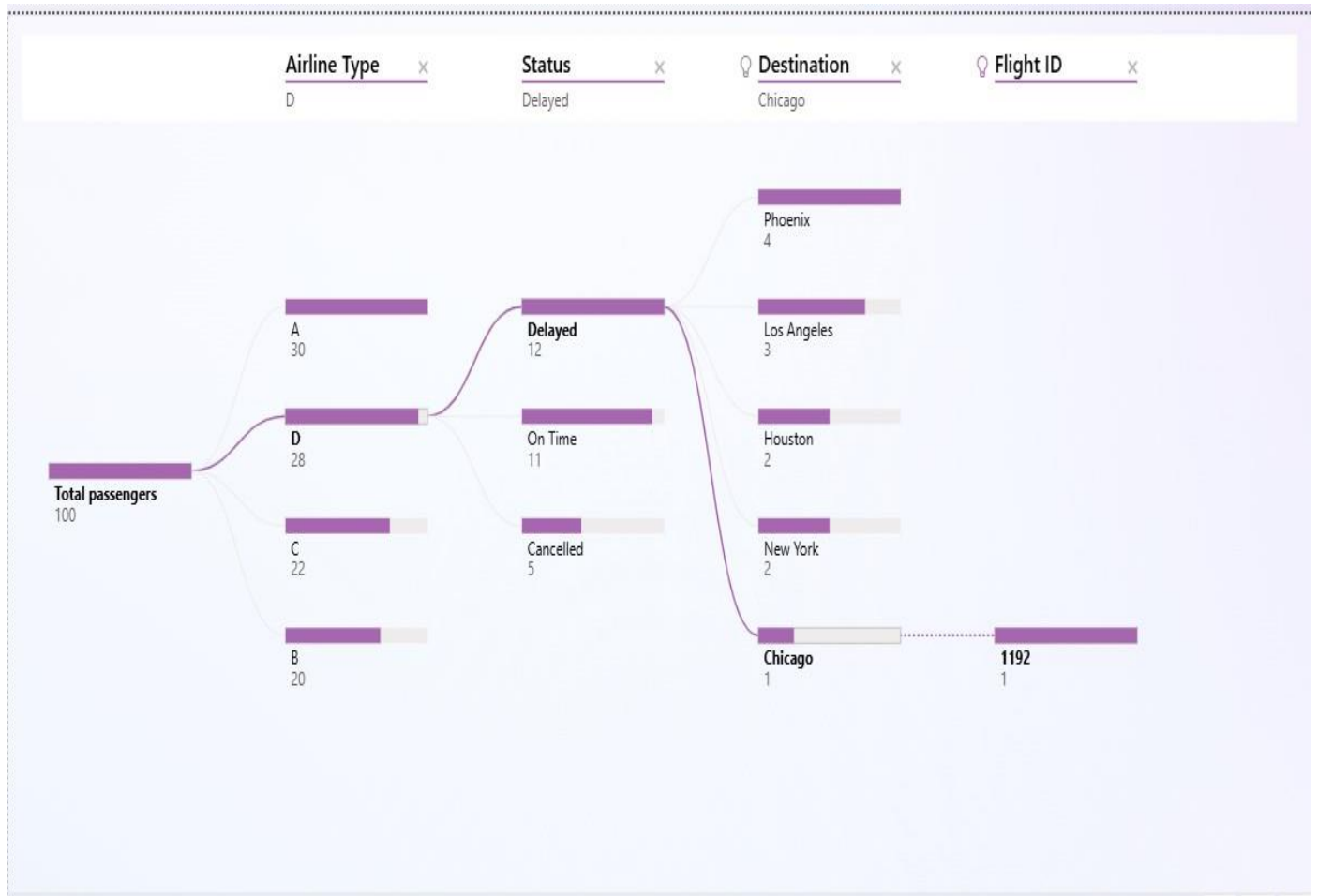
Complete Report Page:



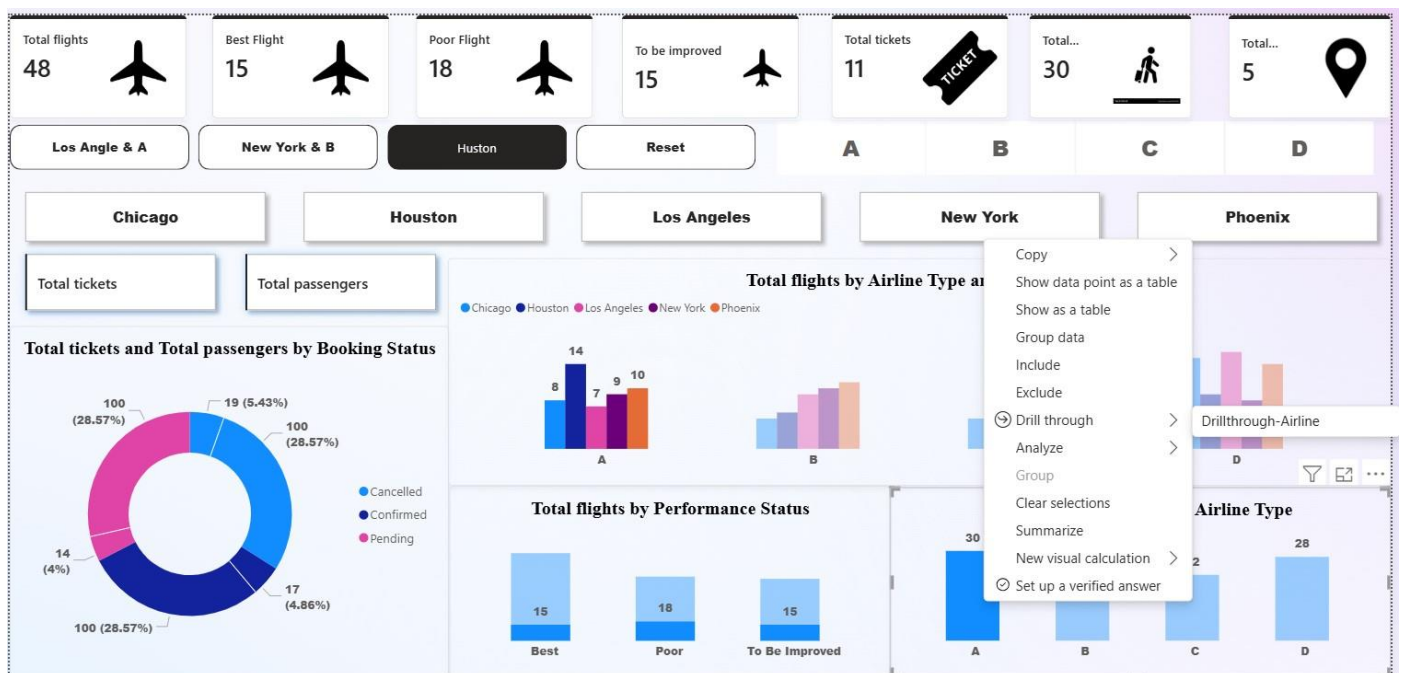
Airline Specific Page:



Decomposition Tree:



Drill Through:



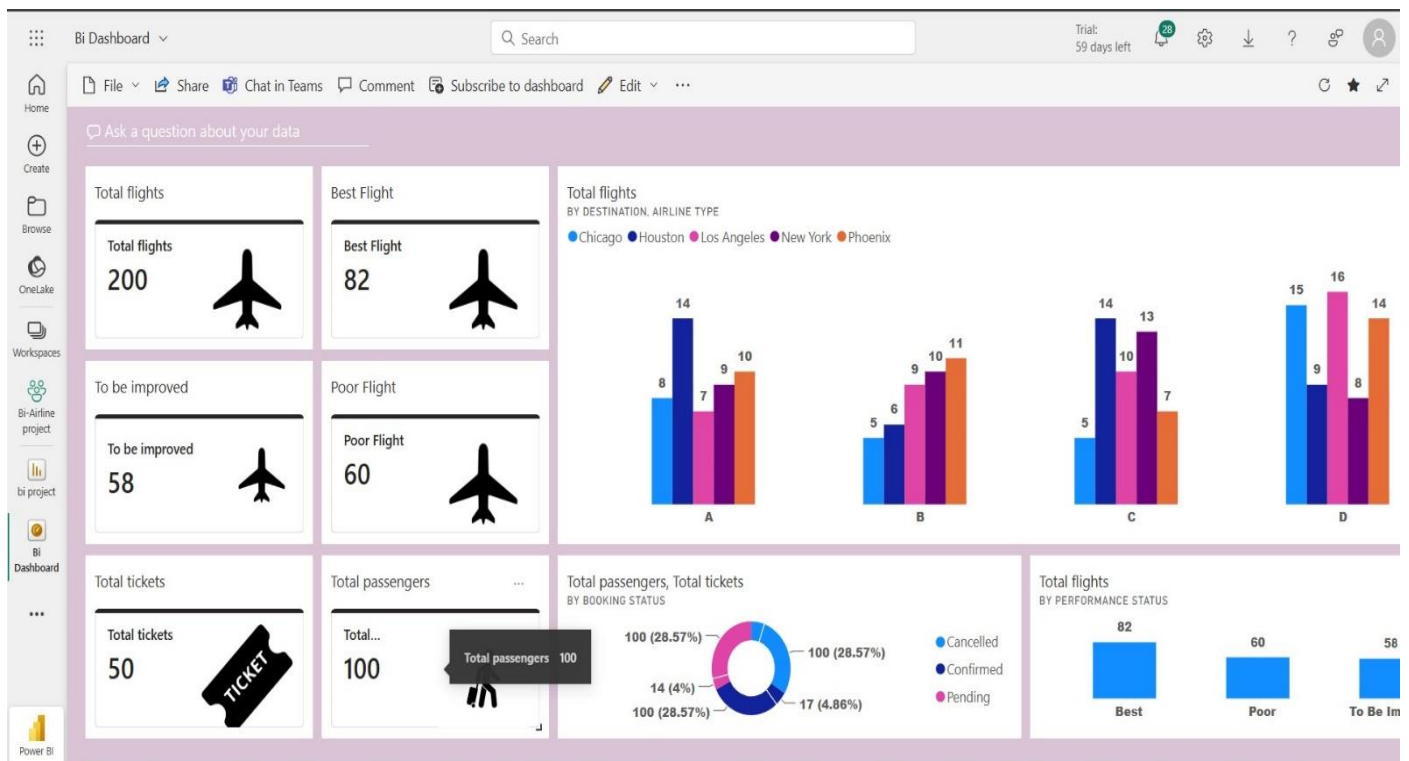
6. Final Dashboard and Power BI Service

A final dashboard was developed in Power BI Desktop and published in Power BI Service.

Key Activities:

Development of a summary dashboard that displayed:

- Total Flights
- Passenger Volume
- Booking Status Distribution



Row Level Security (RLS) Activity:

- RLS was set-up to ensure that the data that the current user could view was limited to only "Airline A"
- A role was created and assigned to a test user on the Power BI Service.

Row-Level Security

Airline - A (1)

Members (1)

People or groups who belong to this role

Enter email addresses

Add

xi mas

Save Cancel

Manage security roles



Create new security roles and use filters to define row-level data restrictions.

Roles

+ New

Airline - A ...

Select tables

- Best Flights
- Flight_Informa...
- Measures (2)
- Parameter
- Passenger_Info...
- Ticket_Informa...

Filter data

+ New Select all Delete Group Ungroup

Show data if All of these rules are true

Column	Condition	Value
Airline Type	Equals	A
+ New		

Save

Close

Scheduled Data Refresh:

- Made configuration for a daily refresh for the dataset at 5:00 PM in Power BI Service

Power BI BI-Airline project Search Trial: 59 days left

Apply Discard

- Gateway and cloud connections
- Data source credentials
- Parameters
- Refresh
 - Time zone
 - (UTC+05:30) Chennai, Kolkata, Mumbai
 - Configure a refresh schedule
 - Define a data refresh schedule to import data from the data source into the semantic model. [Learn more](#)
 - On
 - Refresh frequency: Daily
 - Time: 5:00 PM
 - [Add another time](#)
 - Send refresh failure notifications to

Final Thoughts

This project illustrates the power of Power BI to convert unprocessed airline data into actionable business intelligence. Each data ingestion to deploying the dashboard was done to support real-world analytics objectives — enhance operational transparency and better inform decision-making for airline management.

Thank you

Video Link: https://drive.google.com/file/d/1xl-FfngraPIvN35yDAHmeYiexj_YGZfU/view?usp=sharing