Flight Data Analysis

Your Gateway to Memorable Travel

Project Objective And Scope

Analyze Flight Delays & Performance

- Identify patterns and trends in flight delays, on-time performance across airlines and airports.
- Understand the impact of taxi-in, taxi-out, and cancellations on overall flight schedules.

Optimize Airline & Airport Operations

- Provide insights to improve scheduling, reduce delays, and enhance ground operations.
- Help airlines and airports better manage resources based on delay trends and seasonal flight performance.

Enhance Passenger Experience & Reliability

- Help passengers make informed travel decisions.
- Improve customer satisfaction by reducing unexpected travel disruptions.

Goal Of The Project Market

The final goal of this project is to provide a comprehensive analysis of flight's operational efficiency by utilizing OLAP queries to extract meaningful insights and cost management

- Understand factors causing delays and cancellations.
- Identify patterns in delays based on various aligned factors.
- Airline and airport performance benchmarking.

Software Implementation

Data Processing & Cleaning

Jupyter Notebook – Interactive development environment for analysis.

Python – Data manipulation, cleaning, and handling missing values.

Google BigQuery – Running SQL queries on large datasets efficiently.

ETL & Data Transformation

Google BigQuery – Data transformation, OLAP queries, and efficient data warehousing.

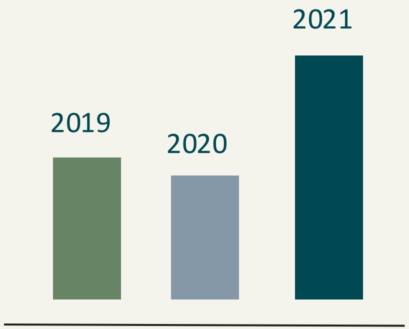
Data Visualization & Analysis

Tableau – Connected **Tableau to BigQuery** for real-time data visualization and analysis.

Star Schema Design

Lucid Chart

Data & Data Source



Data Source: Kaggle

Data Span: 3 years

Flight data: 19 million rows and 61 columns

Data Type: Parquet format

```
FlightDate                           datetime64[ns]
Airline                                    category
Flight_Number_Marketing_Airline               int64
Origin                                     category
Dest                                       category
Cancelled                                      bool
Diverted                                       bool
CRSDepTime                                    int64
DepTime                                     float64
DepDelayMinutes                             float64
OriginAirportID                               int64
OriginCityName                               object
OriginStateName                            category
DestAirportID                                 int64
DestCityName                                 object
DestStateName                              category
TaxiOut                                     float64
TaxiIn                                      float64
CRSArrTime                                    int64
ArrTime                                     float64
ArrDelayMinutes                             float64
AirTime                                     float64
CRSElapsedTime                              float64
ActualElapsedTime                           float64
Distance                                    float64
DayOfWeek                                     int64
DepDel15                                    float64
ArrDel15                                    float64
DepartureDelayGroups                        float64
ArrivalDelayGroups                          float64
dtype: object
```

Data Warehouse Design Overview

- **Data Warehouse :** Google BigQuery
- **Schema Type :** Star Schema
- **Query Language :** BigQuery SQL
- **BI Tool** : Tableau

ETL

EXTRACTION

Data Source

- Flight records in Parquet format from Kaggle.

TRANSFORMATIONS

Data Cleaning & Standardization

- Removed duplicate records.
- Handle Outliers
- Handled missing values (e.g., filling with defaults or removing incomplete rows).
- Standardized date & time formats (converted timestamps to a uniform format).

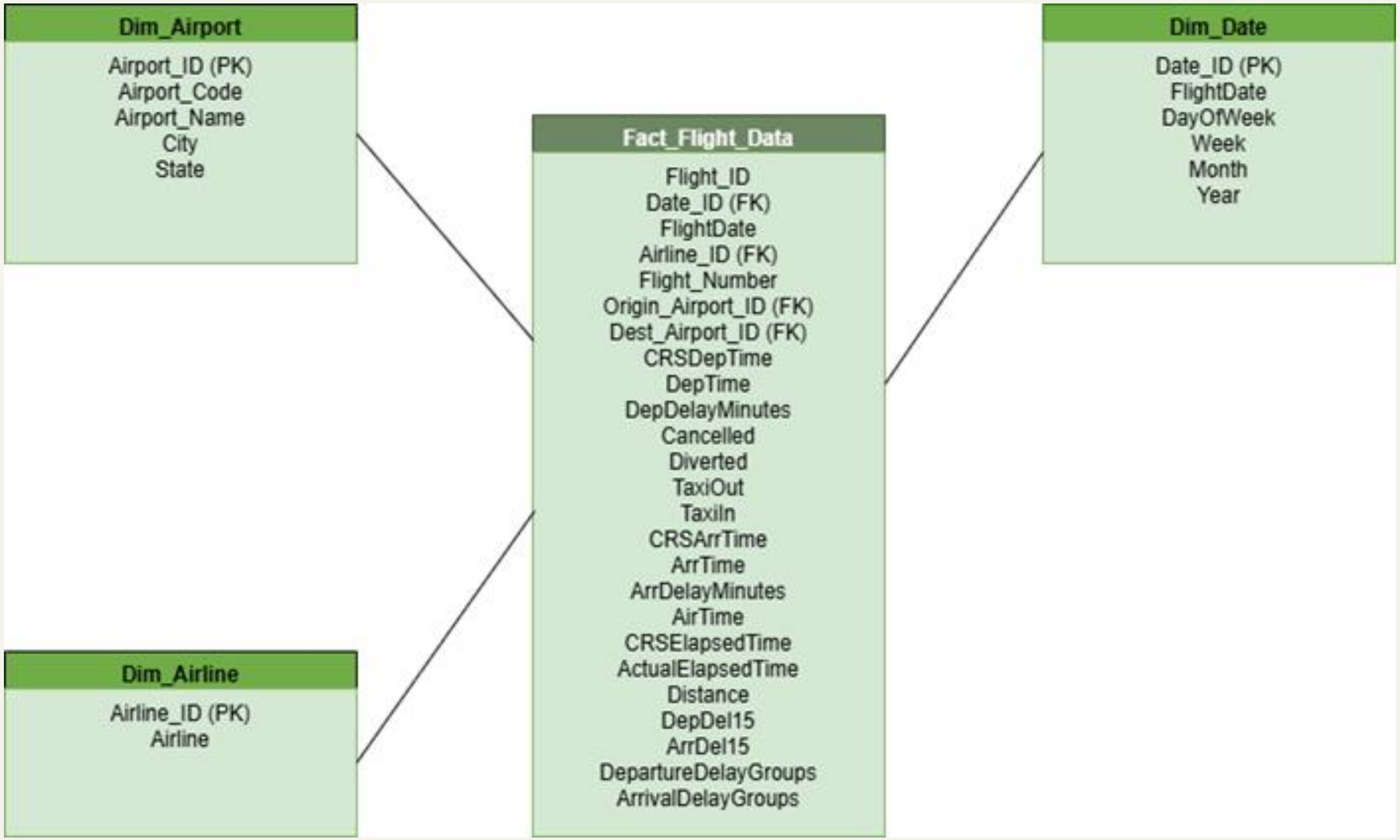
LOADING

Schema Design

Transformed data was loaded into Google BigQuery.

- Created Fact Table and Dimension Tables:
- Aggregate data for OLAP operations.
- Join fact and dimension tables.

Star Schema


Dim_Airport
Airport_ID (PK)
Airport_Code
Airport_Name
City
State
Fact_Flight_Data
Flight_ID
Date_ID (FK)
FlightDate
Airline_ID (FK)
Flight_Number
Origin_Airport_ID (FK)
Dest_Airport_ID (FK)
CRSDepTime
DepTime
DepDelayMinutes
Cancelled
Diverted
TaxiOut
TaxiIn
CRSArrTime
ArrTime
ArrDelayMinutes
AirTime
CRSElapsedTime
ActualElapsedTime
Distance
DepDel15
ArrDel15
DepartureDelayGroups
ArrivalDelayGroups
Dim_Date
Date_ID (PK)
FlightDate
DayOfWeek
Week
Month
Year
Dim_Airline
Airline_ID (PK)
Airline

Fact & Dimension Table

dim_airport

Field name	Type
Airport_ID	STRING
Airport_Code	STRING
Airport_Name	STRING
City	STRING
State	STRING

dim_date

Field name	Type
Date_ID	STRING
FlightDate	DATE
DayOfWeek	INTEGER
Week	INTEGER
Month	INTEGER
Year	INTEGER

dim_airline

Field name	Type
Airline_ID	INTEGER
Airline	STRING

Fact Table

Field name	Type
Flight_ID	STRING
Date_ID	STRING
FlightDate	DATE
Airline_ID	INTEGER
Flight_Number_Marketing_Airline	STRING
Origin_Airport_ID	STRING
Dest_Airport_ID	STRING
CRSDepTime	INTEGER
DepTime	INTEGER
DepDelayMinutes	INTEGER
Cancelled	INTEGER
Diverted	INTEGER
TaxiOut	INTEGER
TaxiIn	INTEGER
CRSArrTime	INTEGER
ArrTime	INTEGER
ArrDelayMinutes	INTEGER
AirTime	INTEGER
CRSElapsedTime	INTEGER
ActualElapsedTime	INTEGER
Distance	FLOAT
DepDel15	INTEGER
ArrDel15	INTEGER
DepartureDelayGroups	INTEGER
ArrivalDelayGroups	INTEGER

Incremental Update of DW

Monthly Update

- New flight records for the latest month (Fact Table)
- New airlines, airports, and dates (Dimension Tables)

Steps

Extracting new or modified records from the source system.

- Comparing against existing data in DW.
- Inserting new records and update existing ones.
- Automating it using Python (ETL),SQL.

Scripts of The Incremental Update

```
INSERT INTO bamboo-medium-450316-m8.flight_data.fact_flight_data (
    Flight_ID, Date_ID, Airline_ID, Origin_Airport_ID, Dest_Airport_ID,
    DepDelayMinutes, ArrDelayMinutes, Cancelled, FlightDate, LastUpdated
)
SELECT
    source.Flight_ID, source.Date_ID, source.Airline_ID,
    source.Origin_Airport_ID, source.Dest_Airport_ID,
    source.DepDelayMinutes, source.ArrDelayMinutes,
    source.Cancelled, source.FlightDate, CURRENT_TIMESTAMP
FROM bamboo-medium-450316-m8.flight_data.staging_fact_flight_data AS source
WHERE EXTRACT(YEAR FROM source.FlightDate) = EXTRACT(YEAR FROM CURRENT_DATE())
  AND EXTRACT(MONTH FROM source.FlightDate) = EXTRACT(MONTH FROM CURRENT_DATE())
ON DUPLICATE KEY UPDATE
    DepDelayMinutes = VALUES(DepDelayMinutes),
    ArrDelayMinutes = VALUES(ArrDelayMinutes),
    Cancelled = VALUES(Cancelled),
    LastUpdated = CURRENT_TIMESTAMP;
```

```
INSERT INTO dim_date (Date_ID, FlightDate, Year, Month, DayOfWeek)
SELECT DISTINCT
    DATE_FORMAT(FlightDate, '%Y%m%d') AS Date_ID,
    FlightDate,
    YEAR(FlightDate),
    MONTH(FlightDate),
    DAYOFWEEK(FlightDate)
FROM staging_fact_flight_data
WHERE EXTRACT(YEAR FROM FlightDate) = EXTRACT(YEAR FROM CURRENT_DATE())
  AND EXTRACT(MONTH FROM FlightDate) = EXTRACT(MONTH FROM CURRENT_DATE())
ON DUPLICATE KEY UPDATE
    Year = VALUES(Year),
    Month = VALUES(Month),
    DayOfWeek = VALUES(DayOfWeek);
```

```
INSERT INTO dim_airline (Airline_ID, Airline)
SELECT DISTINCT Airline_ID, Airline
FROM staging_fact_flight_data
ON DUPLICATE KEY UPDATE
    Airline = VALUES(Airline);

INSERT INTO dim_airport (Airport_ID, Airport_Code, Airport_Name, City, State)
SELECT DISTINCT Origin_Airport_ID AS Airport_ID, Origin AS Airport_Code, OriginCityName AS Airport_Name, OriginStateName AS State
FROM staging_fact_flight_data
ON DUPLICATE KEY UPDATE
    Airport_Code = VALUES(Airport_Code),
    Airport_Name = VALUES(Airport_Name),
    State = VALUES(State);
```

Business Intelligence

	End Users	Use case
01	Executives	High-level reports for strategic decision-making (e.g., yearly flight performance trends)
02	Business Analysts	Data exploration, trend analysis, and predictive modeling
03	Operations Team	Monitoring flight delays, cancellations, and efficiency metrics
04	Marketing Team	Understanding passenger demand and customer preferences

OLAP QUERIES

1. Roll-Up : Aggregating Flight Delays by Year & Airline

- The results will show which airlines experience the highest total departure and arrival delays over different years.
- Helps airlines implement strategies to reduce delays and improve scheduling efficiency.
- Help stakeholders in the airline industry make data-driven decisions on scheduling, fleet management, and resource allocation.

```sql
SELECT
    d.Year,
    a.Airline,
    SUM(f.DepDelayMinutes) AS Total_Departure_Delay,
    SUM(f.ArrDelayMinutes) AS Total_Arrival_Delay
FROM `bamboo-medium-450316-m8.flight_data.fact_flight_data` f
JOIN `bamboo-medium-450316-m8.flight_data.dim_airline` a
    ON f.Airline_ID = a.Airline_ID
JOIN `bamboo-medium-450316-m8.flight_data.dim_date` d
    ON f.Date_ID = d.Date_ID
GROUP BY d.Year, a.Airline
ORDER BY d.Year, Total_Departure_Delay DESC;
```

Row	Year	Airline	Total_Departure_Delay	Total_Arrival_Delay
1	2019	Southwest Airline...	15692786	13517583
2	2019	American Airlines...	13814816	14096412
3	2019	SkyWest Airlines ...	13463873	13684154
4	2019	Delta Air Lines Inc.	10750245	10657128
5	2019	United Air Lines I...	10222473	10581697
6	2019	JetBlue Airways	6420069	6268679
7	2019	Envoy Air	4149527	4633389
8	2019	Republic Airlines	4136433	4599890
9	2019	Comair Inc.	4081732	4106304
10	2019	Mesa Airlines Inc.	3863308	3997880

2. Drill-Down: Taxi-In and Taxi-Out Times by Airport

- Taxi-in and taxi-out times at different airports, which helps in understanding airport congestion, efficiency, and operational delays.

- Helps in deciding which airports need better infrastructure investment, Where airlines should plan buffer times for scheduling.

```sql
SELECT
    a.Airport_Name,
    ROUND(AVG(f.TaxiIn), 2) AS Avg_TaxiIn,
    ROUND(AVG(f.TaxiOut), 2) AS Avg_TaxiOut
FROM `bamboo-medium-450316-m8.flight_data.fact_flight_data` f
JOIN `bamboo-medium-450316-m8.flight_data.dim_airport` a
    ON f.Origin_Airport_ID = CAST(a.Airport_ID AS STRING)
WHERE a.Airport_Name IS NOT NULL
GROUP BY a.Airport_Name
ORDER BY Avg_TaxiOut DESC;
```

Row	Airport_Name	Avg_TaxiIn	Avg_TaxiOut
1	Williston, ND, North Dakota	8.53	23.39
2	New York, NY, New York	7.94	23.07
3	Newark, NJ, New Jersey	7.55	23.01
4	Presque Isle/Houlton, ME, Maine	12.85	22.76
5	Dickinson, ND, North Dakota	10.67	21.52
6	Charlotte, NC, North Carolina	6.21	21.31
7	Hayden, CO, Colorado	9.36	21.18
8	Aspen, CO, Colorado	9.78	21.12
9	Philadelphia, PA, Pennsylvania	7.48	21.11
10	Mammoth Lakes, CA, California	10.94	20.3

3. Dice: On-Time Performance for Flights Over a Certain Distance

- Airlines can use this data to optimize scheduling & reduce delays.
- Travelers can use it to choose airlines with better long-distance punctuality.
- Airports can use this to identify carriers causing congestion & improve operations.

```sql
SELECT
    a.Airline,
    f.Distance,
    COUNT(f.Flight_ID) AS Total_Flights,
    SUM(CASE WHEN f.DepDel15 = 1 THEN 1 ELSE 0 END) AS Delayed_Flights
FROM `bamboo-medium-450316-m8.flight_data.fact_flight_data` f
JOIN `bamboo-medium-450316-m8.flight_data.dim_airline` a
    ON CAST(f.Airline_ID AS INT64) = a.Airline_ID
WHERE f.Distance > 1000
GROUP BY a.Airline, f.Distance
ORDER BY f.Distance DESC;
```

Row	Airline	Distance	Total_Flights	Delayed_Flights
1	United Air Lines Inc.	5812.0	20	0
2	Hawaiian Airlines Inc.	5095.0	935	155
3	Hawaiian Airlines Inc.	4983.0	1493	263
4	Delta Air Lines Inc.	4983.0	28	10
5	United Air Lines Inc.	4962.0	1444	245
6	United Air Lines Inc.	4904.0	200	61
7	United Air Lines Inc.	4817.0	692	104
8	Hawaiian Airlines Inc.	4757.0	201	25
9	American Airlines Inc.	4678.0	476	76
10	Delta Air Lines Inc.	4502.0	1577	248

4. Pivot: Average Arrival Delay by Airline

- Airlines can use this to benchmark their performance against competitors and develop strategies to improve operational efficiency.
- Travelers can make informed choices by selecting airlines with a strong on-time arrival record.
- Airports can manage their ground operations more effectively by understanding which airlines contribute to congestion.

```
SELECT * FROM (
  SELECT
    a.Airline,
    d.Year,
    AVG(f.ArrDelayMinutes) AS Avg_Arrival_Delay
  FROM `bamboo-medium-450316-m8.flight_data.fact_flight_data` f
  JOIN `bamboo-medium-450316-m8.flight_data.dim_airline` a
    ON CAST(f.Airline_ID AS INT64) = a.Airline_ID
  JOIN `bamboo-medium-450316-m8.flight_data.dim_date` d
    ON f.Date_ID = d.Date_ID
  GROUP BY a.Airline, d.Year
)
PIVOT (
  AVG(Avg_Arrival_Delay)
  FOR Year IN (2019, 2020, 2021)
);
```

Row	Airline	_2019	_2020	_2021
1	Allegiant Air	15.55652365405...	13.33111095180...	21.27442264002...
2	Delta Air Lines Inc.	10.78629386908...	6.209070238867...	8.509148803307...
3	Mesa Airlines Inc.	18.11938850893...	10.34442677778...	18.24994985603...
4	Endeavor Air Inc.	14.63693001112...	5.877866220211...	6.729974272142...
5	Horizon Air	8.499155000630...	5.351231500191...	7.465947329919...

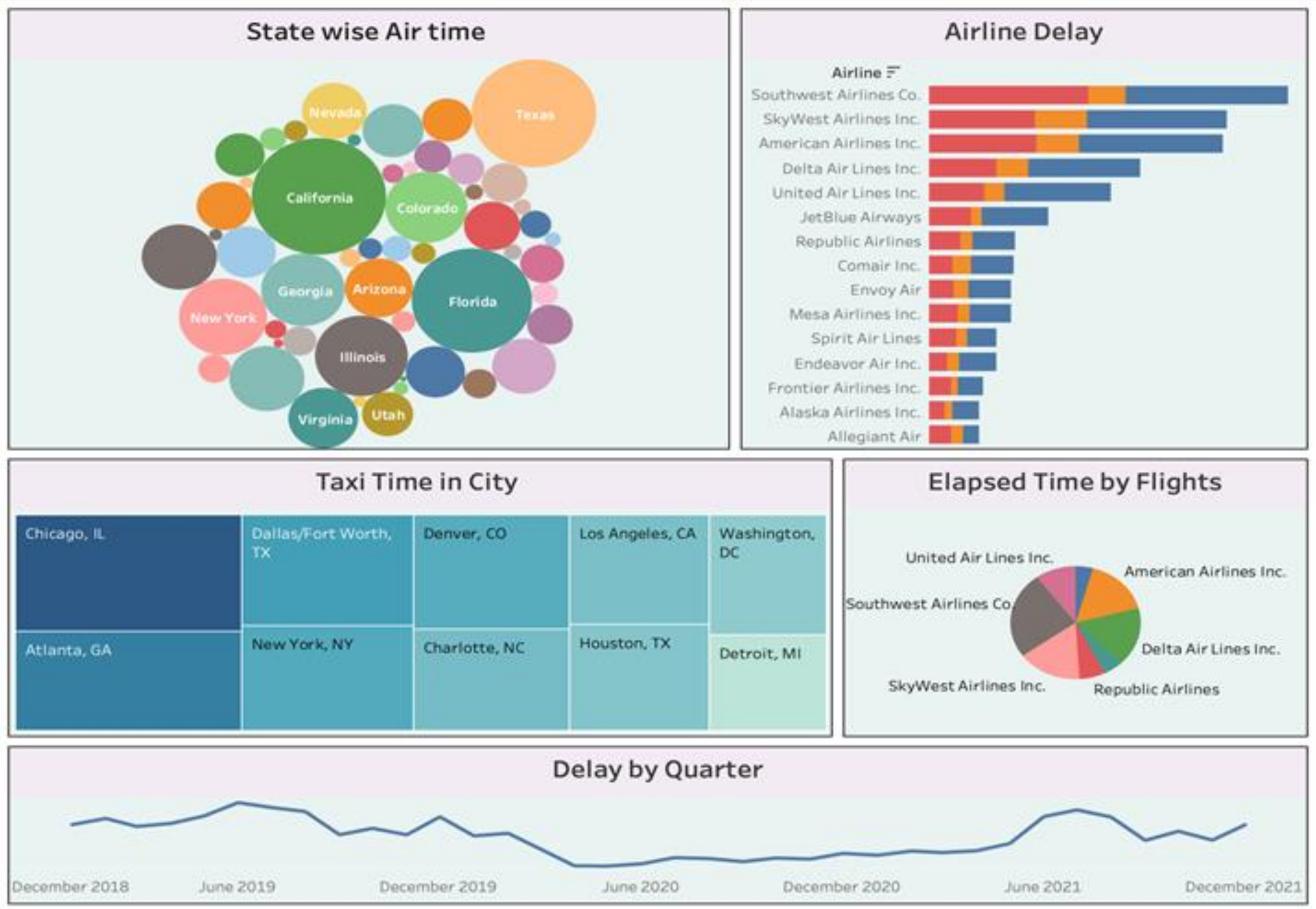
5. Cube: Total Flights by Airline, Month, and Airport

- Airports with high total flights may serve as key hubs, influencing airline scheduling, resource allocation.
- This helps to optimize flight schedules, increase capacity during peak seasons, and reduce costs in off-peak months.

```sql
SELECT
  a.Airline,
  d.Month,
  ap.Airport_Name,
  COUNT(*) AS total_flights
FROM `bamboo-medium-450316-m8.flight_data.fact_flight_data` f
JOIN `bamboo-medium-450316-m8.flight_data.dim_airline` a
    ON CAST(f.Airline_ID AS INT64) = a.Airline_ID
JOIN `bamboo-medium-450316-m8.flight_data.dim_date` d
    ON f.Date_ID = d.Date_ID
JOIN `bamboo-medium-450316-m8.flight_data.dim_airport` ap
    ON CAST(f.Origin_Airport_ID AS INT64) = CAST(ap.Airport_ID AS INT64)
GROUP BY a.Airline, d.Month, ap.Airport_Name
ORDER BY total_flights DESC;
```

Row	Airline	Month	Airport_Name	total_flights
1	Delta Air Lines Inc.	3	Atlanta, GA, Georgia	55518
2	Delta Air Lines Inc.	8	Atlanta, GA, Georgia	52172
3	Delta Air Lines Inc.	1	Atlanta, GA, Georgia	50391
4	Delta Air Lines Inc.	10	Atlanta, GA, Georgia	50238
5	Delta Air Lines Inc.	7	Atlanta, GA, Georgia	49401

Visualization



Summary

Challenges

- Handling missing & duplicate data
- Integrating with Google cloud platform
- Optimizing schema design

Learnings

- Google BigQuery Performance
- Understanding OLAP Operations
- Visualization & Reporting in Tableau

Future Scope

- Real-Time Flight Delay Prediction
- Automating ETL Workflows
- Flight Optimization System

Thank You