Creating Flight Information Data Pipelines

Vivian Do

Stephen Reagin

Mohammad Mahmoudighaznavi

University of San Diego

Shiley-Marcos School of Engineering

February 26, 2023

GitHub Repository Link

Our Github repository can be found at: https://github.com/sfreagin/ADS507_airlines. It includes all instructions to reproduce, deploy, and monitor the pipelines.

Source Datasets

We are looking at flight data, specifically flights that occurred between 2018 and 2022 from four different airports in San Diego (SAN), Seattle (SEA), New York (JFK), and Houston (IAH). The data source is the *anyflights* library which uses an API to download data through R code, having documentation at https://anyflights.netlify.app/reference/get_flights.html and https://github.com/simonpcouch/anyflights which is also the source GitHub repository. We initially store these as CSV files, approximately 600MB in total:

- Flights departing from SAN, SEA, JFK, and IAH (one file per year, 5 files total)
- Weather data at the airports (one file per airport per year, 20 files total)
- Airlines involved (one file per year, 5 files total)

The airline industry plays an important and critical role in the world's transportation sector. Multiple businesses and industries rely on the aviation industry to connect with various parts of the world. But varied factors, including: extreme weather conditions, air traffic control, airport operations, increased traffic volume, etc., have the capability to directly and indirectly affect airline services, leading to delayed flights (McCarthy, 2022). Understanding the effects of such issues beforehand would allow the airline operators to be better prepared for the potential reasons for flight delays in advance. If this data is also available to the consumers and/or passengers, it may help them in planning their journey more efficiently. We selected four airports

from Houston, San Diego, New York, and Seattle with different weather conditions during a five-year model to demonstrate our project objectives.

Pipeline Output

Data pipelines are created to serve both analytical and operational purposes for travelers, airlines, and other consumers like airports. We created 119 views from our base data tables, providing useful results through table joins, aggregations, and making necessary calculations to produce practical parameters.

The views can be differentiated as followed:

- Daily flight delays and weather information for each airport per year.
- Specified holiday flights with their average delay time for each year.
- Total delays based on destination separated in each year.
- Monthly average delays for each carrier separated by year and airport.
- Total yearly flights and their delay for each carrier.
- Scheduled total flights with their delay based on time of the day for all origin airports.
- Seasonal delay time for each airport.
- Yearly delays for each origin and destination based on flight time with their average departure and arrival delay time.

The output for each view could be used by airlines, travelers, or other consumers to have a better understanding of flights and delays. For example, travelers can find out what time of day has the most delays, or whether a particular holiday season is a good time to travel. From airline perspectives, they can see the aggregations of flight delays and compare it to other carriers. This information helps them make plans to improve their services.

Architecture Diagram

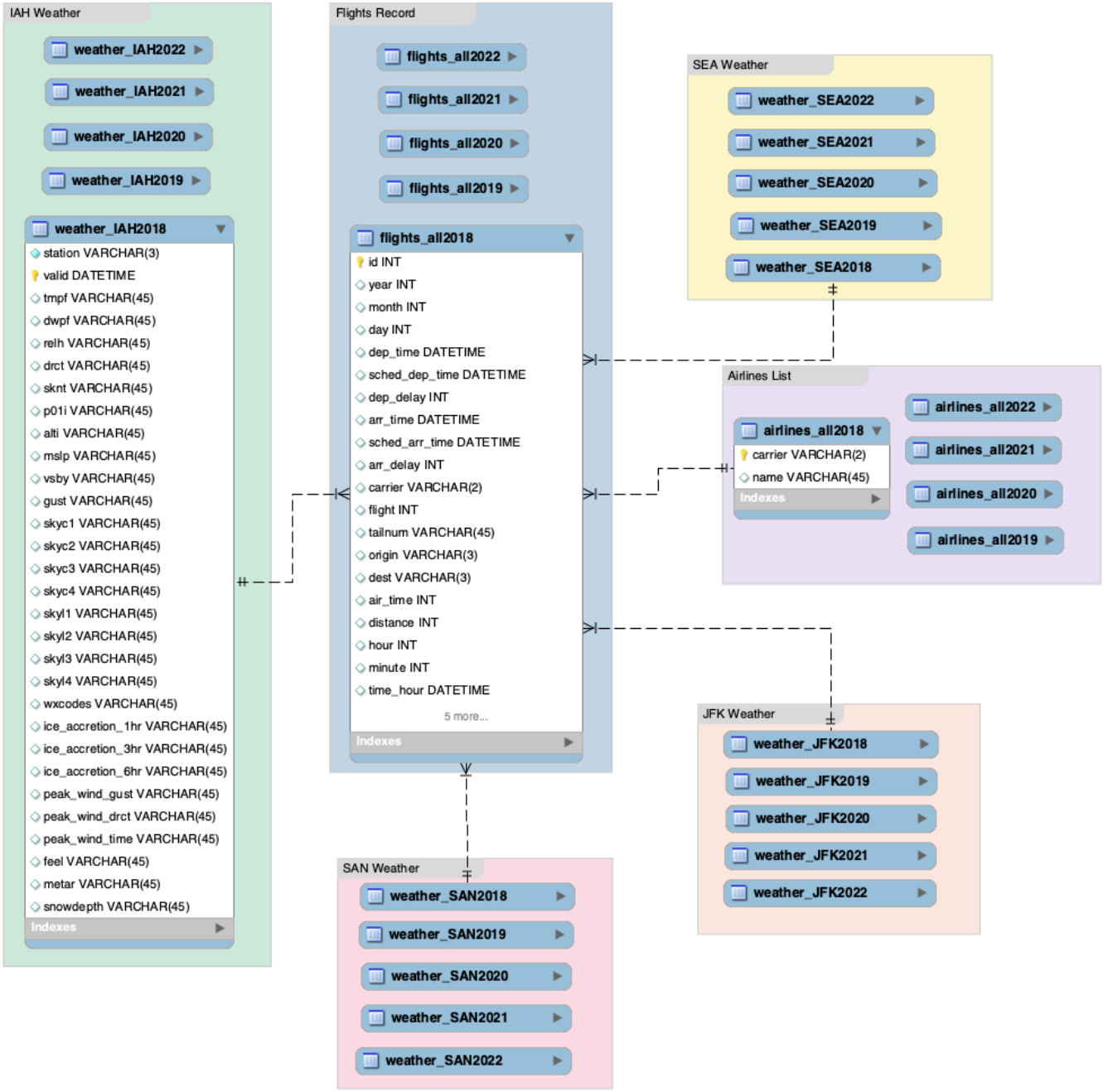
For each year between 2018-2022, the database contains three types of tables:

- **airlines_all20[XX]:** a list of airlines operating from any of the four origin airports for the year, containing the unique two-letter carrier code (‘carrier’) and company (‘name’).
- **flights_all20]XX]:** a record of flights departing from all origin airports for the year. Each flight is uniquely identified by an ID, and contains information regarding scheduled arrival and departure times, actual arrival and departure times, and other flight logistics. See Table 1 in Appendix A for a full list of fields and definitions.
- **weather_[XYZ]20[XX]:** recorded weather measurements, at five minute intervals, for each airport. Tables for each origin airport contain the same fields, except for ‘station’, which pertains to the three-character site identifier for that particular airport. See Table 2 in Appendix B for a full list of fields and definitions.

For ease of clarity, the architecture diagram in Figure 1 does not depict relationships for all tables in the database. Rather, it only shows the relationships between airlines, weather at each origin airport, and flights for the year 2018. This can be used as a reference for tables in subsequent years.

Figure 1

Architecture Diagram



Will the system scale as the dataset size grows?

The system is capable of performing well when it comes to bigger datasets. The only concern here is the processing time which could be longer than what we have right now. For example, if the database size increases to all airports around the country or even the world, to make sure the system remains performant and usable for its users, we might need to add more servers in order to process our larger stream of data.

That said, the data for years 2018 through 2021 is fixed for the airports under consideration and will not increase further. Any additional airports or years of information will require new base table creation as further described below.

Is the system secure?

The data pipeline that we have created is intended to be readily accessible and reproducible for the general public. Additionally, the data is open-sourced and does not contain personal identifiable information of any individual, so the risk of data privacy issues is low. However, we have taken steps to increase security of the database, mainly around the people and processes involved.

Firstly, the credentials of all contributors are protected using the Python package 'getpass'. This package hides the MySQL server password of the contributor when they are connecting to the database from a Jupyter Notebook. Secondly, the database has been backed up in each contributor's local system. Should the data become compromised (e.g the Github file becomes corrupted), the database can be recovered from any of the three locations.

Is the system extensible?

As currently constructed, the database will not extend to new airports or years without new SQL code to create additional base tables. In addition, the **weather_[XYZ].csv** files require manual editing to remove the first several rows of data, which contain comments that cause import errors in both SQL and Python. To enhance extensibility, we would ideally have a database design which automates the following process:

1) Capture data through API calls
2) Pre-process the original CSV files for SQL readability
3) Create base tables from the pre-processed CSV files
4) Design appropriate views from the newly-created base tables

That said, the database *is* extensible to new business problems and other analytical projects, such as processing data for AI/ML modeling, business dashboards, user interfaces, and many other use cases. We have thus far demonstrated pipeline outputs for key questions but also see the potential for answering many other problems through creation of new views or other manipulations of the underlying base tables as currently constructed.

References

Bureau of Transportation Statistics. (n.d.). https://www.bts.gov/

Couch, S. (n.d.). *Anyflights Documentation.* Retrieved February 26, 2023 from: https://anyflights.netlify.app/reference/anyflights.html

Hotle, S., & Mumbower, S. (2023). The impact of COVID-19 on domestic U.S. air travel operations and commercial airport service. *ScienceDirect*, 1-5. Retrieved from ScienceDirect: https://www.sciencedirect.com/science/article/pii/S2590198220301883

McCarthy, D. (2022, April 08). *What Are the Most Common Reasons For Flight Delays in the U.S.?* Retrieved from travelmarketreport: https://www.travelmarketreport.com/News/articles/What-Are-the-Most-Common-Reasons-For-Flight-Delays-in-the-US

Reis, J. & Housley, M. (2022). *Fundamentals of data engineering*. O'Reilly.

Appendix A

Flight Table Fields and Definitions

Fields

station	Three or four character site identifier
valid	Timestamp of the observation
tmpf	Air Temperature in Fahrenheit, typically @ 2 meters
dwpf	Dew Point Temperature in Fahrenheit, typically @ 2 meters
relh	Relative Humidity in %
drct	Wind Direction in degrees from *true* north
sknt	Wind Speed in knots
p01i	One hour precipitation for the period from the observation time to the time of the previous hourly precipitation reset. This varies slightly by site. Values are in inches. This value may or may not contain frozen precipitation melted by some device on the sensor or estimated by some other means. Unfortunately, we do not know of an authoritative database denoting which station has which sensor
alti	Pressure altimeter in inches
mslp	Sea Level Pressure in millibar
vsby	Visibility in miles
gust	Wind Gust in knots
skyc1	Sky Level 1 Coverage
skyc2	Sky Level 2 Coverage
skyc3	Sky Level 3 Coverage
skyc4	Sky Level 4 Coverage
skyl1	Sky Level 1 Altitude in feet

skyl2	Sky Level 2 Altitude in feet
skyl3	Sky Level 3 Altitude in feet
skyl4	Sky Level 4 Altitude in feet
wxcodes	Present Weather Codes (space separated)
ice_accretion_1hr	Ice Accretion over 1 Hour (inches)
ice_accretion_3hr	Ice Accretion over 3 Hours (inches)
ice_accretion_6hr	Ice Accretion over 6 Hours (inches)
peak_wind_gust	Peak Wind Gust (from PK WND METAR remark) (knots)
peak_wind_drct	Peak Wind Gust Direction (from PK WND METAR remark) (deg)
peak_wind_time	Peak Wind Gust Time (from PK WND METAR remark)
feel	Apparent Temperature (Wind Chill or Heat Index) in Fahrenheit
metar	Unprocessed reported observation in METAR format

Note. From Iowa State University, Iowa Environmental Mesonet "ASOS-AWOS-METAR Data Download". Retrieved February 22, 2023, from https://mesonet.agron.iastate.edu/request/download.phtml.

Appendix B

Weather Table Fields and Definitions

Fields

id	unique identifier for each flight record
year	year of flight
month	month of flight
day	day of flight
dep_time	time of flight departure from origin airport
sched_dep_time	scheduled departure time from origin airport
dep_delay	amount of time (in minutes) that the flight was delayed in its departure; a negative value denotes that a flight departed before its scheduled time.
arr_time	time of flight arrival to destination airport
sched_arr_time	scheduled arrival time to destination airport
arr_delay	amount of time (in minutes) that the flight was delayed in its arrival; a negative value denotes that a flight arrived before its scheduled time.
carrier	unique two-letter airline code
flight	flight number
tailnum	alphanumeric code used to identify the aircraft
origin	origin airport (IAH, JFK, SAN, or SEA)
dest	destination airport
air_time	amount of time the aircraft was in the air (in minutes)
distance	number of miles traveled
hour	hour of the scheduled departure time

minute	minute of the scheduled departure time
time_hour	scheduled departure time in datetime format, rounded by the hour