Monash Data Bootcamp

ETL Project Report

01/08/2022

ContributoRS:

Andrew King

Paul Wang

Dicle Demirkol (Denise)

# Summary

We chose to work on Australian aviation data including airports, routes and flight data for this project. This report summarizes;

* The steps to be able to run the programs, and extract the data.
* Extraction details, our data sources, file formats.
* Transformation details, how we processed each data file/frame.
* Load details. Why we chose SQL database, and our decisions.

# Activation Steps

1) Run 0\_Table Creation Script.sql using the preferred SQL database tool, to create the database objects.

2) Run 2\_Australia\_AirTraffic.ipynb as a Jupyter Notebook file to import, modify and extract the data to the database.

3) Run 1\_Queries.sql using the preferred SQL database tool, to review the results.

# EXTRACT

The following table shows the data sources we selected, the source file formats and their original location.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| File Source | File Type | File name | Link | Label |
| Resources | CSV | “list-of-airports-in-australia-hxl-tags-1.csv” | https://data.world/ourairports/6fee2ce0-5e8e-4138-a794-f5e9052f38c6/workspace/file?filename=list-of-airports-in-australia-hxl-tags-1.csv | Airport\_data |
| Resources | XLSX | “TopRoutesJuly2014May2022.xlsx” | https://data.world/tobybellwood/domestic-airlines-top-routes | Route\_data |
| Resources | JSON | “worldcities.json” | https://simplemaps.com/data/world-cities | City\_data |
| Wikipedia | HTML | Scraped from Wikipedia | https://en.wikipedia.org/wiki/States\_and\_territories\_of\_Australia | States\_data |
| Resources | XLSX | “ardd\_fatalities\_jun2022.xlsx” | https://www.bitre.gov.au/statistics/safety/fatal\_road\_crash\_database | Fatalities\_data |
| API | JSON | - | https://app.goflightlabs.com/flights? | Flight\_data |

# TRANSFORM

## Airport\_data:

* Selected relevant columns
* Renamed those columns
* Converted iso\_region data to Aus. state code using string functions (e.g., AU-NSW to NSW)

## Route\_data:

* Chose relevant columns using range function
* Renamed the columns
* Dropped columns with incomplete data
* Converted all data to the integer datatype
* We used the lambda and calendar functions to create an additional column with the full month names, rather than their abbreviations (e.g., 1 became January)

## City\_data:

* Filtered the cities to only include those from Australia
* Converted the city strings to all be upper case for uniformity of data
* Renamed the admin column

## States\_data

* Use Beautiful Soup to scrape a html version of the wiki page
* Selected relevant columns
* Renamed those columns
* Created and merged two dataframes to capture the full states / territories list
* Realigned and renamed columns
* Joined dataframes with City\_data so as to use state\_code as the key value

## Fatalities\_data

* Skipped several rows that were headers or lacked data
* Renamed columns
* We used the lambda and calendar functions to create an additional column with the full month names, rather than their abbreviations (e.g., 1 became January)
* Converted the state strings to all be upper case

## Flight\_data

* As we were limited to making 100 API calls per month, we focused on the large airports
* Filtered out irrelevant JSON response data fields
* Responses were captured in lists and converted to a dataframes

# LOAD

We chose a SQL database (PGAdmin), rather than MongoDB, as the data was structured. All dataframes created above were then loaded into SQL table format. In the case of City\_data we chose to use state\_code and iata\_codes to link the tables so as to avoid any spelling, spaces and naming inconsistencies.