US Flights Case Study

# Scenario

A client has provided you with a flat file export containing data about domestic flights in the United States. The analytics consultant that you are working with needs the file loaded into a database and appropriately modeled for use with business intelligence tools. In this case, that tool is Tableau.

## Goal

The goal of this case is to assess your ability to:

1. Work with unfamiliar data and quickly gain familiarity with that data.
2. Load the data from a source into a target database
3. Transform the data into a normalized, dimensional data model with appropriate data types
4. Think critically about preparing data for business users and analytics consultants
5. Present issues with data quality that do not have obvious solutions and solve issues that do have more straightforward solutions

## Data

The data is provided as a compressed flat file. The file is encoded in UTF-8 and is delimited by the pipe character. Column headers are provided on the first line.

**To assist in automated grading, please don’t change the names of any columns in the data.** For example, two columns in the source data are named TRANSACTIONID and CANCELLED. They should be named TRANSACTIONID and CANCELLED in your view, VW\_FLIGHTS. In the example below, we ask you to create a column DISTANCEGROUP and it should be named DISTANCEGROUP in all tables and all views. There should be no underscores or spaces added.

## Case Requirements

1. Load the provided data into the PostgreSQL instance using the provided credentials.
2. Create and load one fact table named FACT\_FLIGHTS to contain data about the flights.
3. Create and load appropriate dimension table(s) named DIM\_\*.
4. Create a view named VW\_FLIGHTS that joins your fact table to your dimension tables and returns columns useful for analysis. Please see the “View” section for more information.
5. Prepare a 10 to 15-minute presentation to show your work, discuss your approach and any data quality issues that were encountered and your resolution to these issues.

# Additional Details and Instructions

## Fact Table

1. Create an additional column named DISTANCEGROUP that bins the distance values into groups in 100-mile increments. Example: 94 miles is 0-100 miles. 274 miles is 201-300 miles.
2. Create an additional column named DEPDELAYGT15 that indicates (0/1) if the departure delay in minutes (DEPDELAY) is greater than 15.
3. Create an additional column named NEXTDAYARR that indicates (0/1) if the flight arrival time (ARRTIME) is the next day after the departure time (DEPTIME).
4. Choose appropriate data types and perform conversions to load the data from the source into these types.
5. Fix obviously bad data when encountered, if possible. Note these instances.

## Dimension Table(s)

1. Create at least one dimension table and load it from the source data.
2. Use your judgment about what columns from the source data should end up in the dimension tables. Be prepared to explain your decisions.
3. Clean up the AIRLINENAME column by removing the airline code from it.
4. Clean up the ORIGAIRPORTNAME and DESTAIRPORTNAME columns by removing the concatenated city and state.
5. Fix obviously bad data when encountered, if possible. Note these instances.

## View

Your final view (VW\_FLIGHTS) should contain all columns you deem useful for analysis.

Please also make sure your view (VW\_FLIGHTS) includes at least each of these columns and remember, as stated above, to not change any of the column names:

* TRANSACTIONID
* DISTANCEGROUP
* DEPDELAYGT15
* NEXTDAYARR
* AIRLINENAME
* ORIGAIRPORTNAME
* DESTAIRPORTNAME

Additionally, if you find and repair any other data issues, please make sure to include those columns in the final view.