ETL Project - AFL Games Status 2018-2020

# Technical Report

Group 8: Phuong Tieu – Angela Liu – Mohammed Nasser – Claudia Sandoval

## Project Proposal

AFL is an important subject in Australians sport since its inception and the quantity of fans are growing steadily as years past. AFL teams and players performance is the first step that fans (current and new ones) look when competing on the yearly seasons. Team 8 will produce an AFL Game DB that will transform the raw data on performance of teams and players so fans can easily query and find relational information about their favourite clubs.

## Pre- Processing

### The sources of data

Data was obtained from the online public dataset:

* Kaggle datasets (<https://www.kaggle.com/>) filtering to find the correct and available data sources for our project.
* AFL Tables (<https://afltables.com/afl/afl_index.html>)
* Austadiums (<https://www.austadiums.com/>)

## Scope of the process:

The source of data we are proposing for the project cover the period 2018-2020, period during which the data is complete. Open-source data has the limitation of accessing historic data only.

# Extraction – Steps 1 to 5

For the extraction process, four different datasets were used from the public platforms which led the team to the AFL information. The following dataset were used:

* Games.csv – from this file we extracted the games results
* Stats.xlsx – from this file we extracted the player’s performance
* AFL Stadiums.xlsx (workbook with 2 xlsx sheets:
  + Stadiums – gave us the stadiums capacity where the games were played.
  + Venue data gave the availability of the stadiums.
  + We needed to join these 2 worksheets to arrange the DB, this task included another layer of cleaning in order to formatting the name.
* AFL team\_venues.csv – from it we extracted the team’s and venue’s names.

The fields of interest which were extracted using the Python language to convert to four data frames in Pandas, were:

* AFL game results,
* AFL players’ performance,
* AFL Venue In-use information,
* AFL stadium location and capacity information,
* AFL team venues (teams and stadiums information)

An **ERD diagram** was developed after evaluating the data columns information and its correlation. From that, the team created seven tables connecting them through primary keys. The diagram has been filed in the repository.

# Transformation – Steps 1 to 7

To transform the public data and use it in our study we performed the following:

* Used Pandas functions in Jupyter Notebook to load all CSV and Excel files.
* Reviewed the files and transformed into data frames.
* Removed the operator’s column and the address column due to missing information which was not relevant to the focus of this study.
* Serial ID were assigned to each data frame to link the tables and develop a relational database.
* Identified duplicates by doing an inner merge on the incident id column across all three data sets.
* We sorted the data in descending order so we could visually see which team had the highest numbers.

## Manipulate Data Frames using Pandas

We transformed the four data frames to 7 tables and connecting each table with different id. The following steps were

1. Player table: contains player, first name, last name:

* Create a filtered data frame for specific columns: first and last name
* Split the first and last name from full name column using split function, and give values to the new column
* Rename the column headers to match the ERD table
* Drop the duplicates and setting the index
* Call the data frame to check the header
* Using the same procedure to create the other 6 tables:

1. Team table: contains team\_id, name(team name), stadium\_id

* Besides the same procedure as city table with pandas, the hard part is to create the city\_id
* Using the range and length to look for the length of the team\_transformed data frame’s row, and set the auto-increment id as the value of the team\_id; the length is counting + 1, and the range starts from 1;

1. City table: contains city\_id, name (city name), and state
2. Stadium table: contains: stadium\_id.name(stadium name),city\_id, start year, end year, capacity ,active-ind

* Stadium table combines two different sources of data, the stadium worksheet and venue worksheet.
* Each source of data gives different name spelling of the same stadium, we replace the other one first using replace function. The purpose of using different join methods was to bring more availability to the data.
* Then set the data frame using set function and get unique names with unique. union to join venue name, stadium name, and venue name together and rename the column
* Split the year information from in-use column and merge the stadium\_name , venue together
* Outer merge (inclusively merge) the stadium table with the venue, stadium data frame, and using if function to set boolean value: if the stadium is active or not , as if the end year is over 2022
* We would like to keep most of the stadium data available first even venue\_df and stadium\_df miss match with each other where some of the city\_id is not given for stadium in the other, and venue in\_use information is not given in this dafa frame.
* And in the end we can dropNa for unwanted information.
* The rest of table is applying the same data cleaning, filtering, aggregating process

1. Player Status table: contains game\_id, player\_id, team\_id, rebond, inside\_50s, clearance, contested possessions, performance
2. Game table contains: game\_id, year, round, date, start time, stadium id, attendance, home team id, home team score, away team id, away team score, rain fall.
3. TVS table: team id, TVS, yr

**Total Vital Statistics (TVS)**

To calculate the TVS - the four performance statistics that impact the outcome of the game are totalled. This has been done for 1 whole year. (It can be done monthly or on a weekly basis to predict the outcome of each game). The four performance stats are Contested Possessions, Rebounds, Clearances, and Inside 50’s.

We could adjust (add weighting) until we are closer to the actual result for the year, because we are comparing our calculations to last year’s AFL ladder (or whatever year we are doing the calculation on. We could keep adjusting until we are extremely close to the actual result by including other performance statistics that may have impact.

# Load

After we pulled in the CSV files and loaded them into the data frames, we did an initial connection to the Postgres database using PG admin to store our original clean data under **AFLGames\_db.** We used the quick database website to create the initial table schema that got loaded into the Postgres database that generated the first set of tables. The criteria for the use of PostgreSQL were based on the type of raw data we found for this project. Most of the tables had one or more identifier that we used as common ground to combine the information in a relational database.

After running the queries and created the new tables with only the relevant information we reconnected to the database and generated additional tables for the data frames

# Limitations and others

Issues in source data:

* Data scope, source has data from 2012 to 18-Mar-2021, our project had selected data from 2018 onwards
* Some stadium names, and team names have different version in different data sources, we need to check if they are within our data scope, if not, we replaced and consolidated the data.

Limitation in new data tables, due to time constraint, the following scenario happen:

* ‘team’ table: stadiums column stores one or two stadium names, no further normalisation.
* ‘city’ table: state column is varchar(50), no further normalization..
* ‘stadium’ table: some rows have start\_year, end\_year columns with null value because data in source data file are null, the null value had led the active\_ind column flags false.

ERD diagram: some columns have ‘?’ show at the end which indicate they are nullable columns.