ETL Project AFL Games 2018-2020

# Technical Report

Team: Phuong – Angela – Mohammed – Claudia

## 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 from the Kaggle datasets (<https://www.kaggle.com/>) filtering to find the correct and available data sources for our project.

The process modelling included the steps detailed in Figure 1. Each step of the process has been developed in the body of the report.

**Figure 1. Process Modelling**

### Graphical user interface Description automatically generated with medium confidence

The biggest advantage to this setup is that transformation and data modelling happen in the analytics database, in PostgreSQL. This gives greater control over how the target work with it, in a common language they all understand.

# Extraction

For the extraction process, we used four different datasets from the public platform Kaggle which led us to the AFL Websites. The following dataset were used:

* Teamstats 2012-2021.csv
* Games.csv
* Stadium Status.xlsx
* AFTl Stadiums.xlsx

The data in the files included the following information:

* Games.csv:
* Location
* Players performance

The fields of interest included the following:

* Teams’ performance statistics
* Player’s performance statistics
* Location
* Season/s

**Some validations are done during Extraction:**

* Reconcile records with the source data
* Make sure that no spam/unwanted data loaded
* Data type check
* Remove all types of duplicate/fragmented data
* Check whether all the keys are in place or not

# Transformation

Data extracted from source server is raw and not usable in its original form. Therefore, it needs to be cleansed, mapped, and transformed. In fact, this is the key step where ETL process adds value and changes data such that insightful reports can be generated.

It is one of the important ETL concepts where you apply a set of functions on extracted data. Data that does not require any transformation is called as **direct move** or **pass-through data**.

In transformation step, you can perform customized operations on data. For instance, if the first name and the last name in a table is in different columns. It is possible to concatenate them before loading.

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

* Used Pandas functions in Jupyter Notebook to load all three CSV 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.
* Identified duplicates by doing an inner merge on the incident id column across all three data sets.
* Created queries to address our hypothesis by grouping the data by state and getting the sum of the number of people killed and the number of people injured. We sorted the data in descending order so we could visually see which state had the highest numbers.

(Screenshots of tables)

**Following are Data Integrity Problems:**

1. Different spelling of the same person like Jon, John, etc.
2. There may be a case that different account numbers are generated by various applications for the same customer.
3. In some data required files remains blank
4. Invalid product collected at POS as manual entry can lead to mistakes.

**Validations are done during this stage**

* Filtering – Select only certain columns to load
* Using rules and lookup tables for Data standardization
* Character Set Conversion and encoding handling
* Conversion of Units of Measurements like Date Time Conversion, currency conversions, numerical conversions, etc.
* Data threshold validation check. For example, age cannot be more than two digits.
* Data flow validation from the staging area to the intermediate tables.
* Required fields should not be left blank.
* Cleaning ( for example, mapping NULL to 0 or Gender Male to “M” and Female to “F” etc.)
* Split a column into multiples and merging multiple columns into a single column.
* Using any complex data validation (e.g., if the first two columns in a row are empty then it automatically reject the row from processing)

# Load

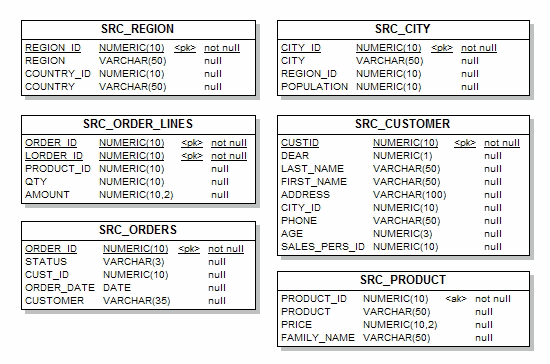
Loading data into the target data warehouse database is the last step of the ETL process. In a typical Data warehouse, huge volume of data needs to be loaded in a relatively short period (nights). Hence, load process should be optimized for performance.

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 sets. 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. 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.

Screenshot of the Database screen

Screenshot of each of the table developed

Schema Diagram



**Load verification**

* Ensure that the key field data is neither missing nor null.
* Test modeling views based on the target tables.
* Check that combined values and calculated measures.
* Data checks in dimension table as well as history table.
* Check the BI reports on the loaded fact and dimension table.

# Summary

There were some limitations to our findings due to the data available (must be completed)

## Integration Challenges

* Creation of integration interfaces to move and transform data

## Constrains

* Handling multiple source formats (csv / xlsx / flat files). To pull in data from diverse sources, your process needs to be able to handle a variety of data formats.
* Scalability: All components of an ETL process should scale to support arbitrarily large throughput.