Extraction, Transformation, and Load (ETL) Report

Authors:

Bryan Barrows

Harinee Madhusudhan

Sandino Tirso

Vivek Mohan

Table of Contents

[Revision History 1](#_Toc528972462)

[1. Introduction 1](#_Toc528972463)

[1.1 Purpose 1](#_Toc528972464)

[1.2 Scope of Initiative 1](#_Toc528972465)

[2. Data Extraction 1](#_Toc528972466)

[2.1 Data Sources 1](#_Toc528972467)

[2.2 Source Data files 1](#_Toc528972468)

[3. Data TransFormation 1](#_Toc528972469)

[3.1 Import Data files 1](#_Toc528972470)

[3.2 Data Cleaning 1](#_Toc528972471)

[3.3 Type conversion 1](#_Toc528972472)

[4. Loading into database 1](#_Toc528972473)

[4.1 Type of Database 1](#_Toc528972474)

[4.2 ORM (SQLAlchemy) 1](#_Toc528972475)

[4.3 Final DB schema including EER Diagram 1](#_Toc528972476)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mm/yy> | <x.x> | <details> | <name> |
| 11/03/2018 | 1.0 | Initial Draft | Team |

# Introduction

## Purpose

The purpose of the Extraction, Transformation, and Load (ETL) Specification Document is to capture details that pertain specifically to ETL development to be used by the developer as an aid in ETL development.

## Scope of Initiative

The scope of this project was to create an MySQL database of American universities ranking and salary data based on multiple heterogenous data sources that were available online. This database has been prepared to be used by student counsellors that are looking to build university recommendation systems or researchers looking in relationship between university rankings and salaries.

# Data Extraction

## Data Sources

The project relied on two primary sources of data:

* The Wall Street Journal ([www.wsj.com](http://www.wsj.com)) college salaries by region which was available as a **.CSV** file on the Kaggle ([www.kaggle.com](http://www.kaggle.com)) website.

<https://www.kaggle.com/wsj/college-salaries#salaries-by-college-type.csv>

* The US News ([www.usnews.com](http://www.usnews.com)) university statistics data that was available for download as a **JSON** file on the Kaggle ([www.kaggle.com](http://www.kaggle.com)) website

<https://www.kaggle.com/theriley106/university-statistics>

* A secondary .csv file which contained the State names and State abbreviations was also used to obtain the abbreviation for each state

## Source Data files

Below are the fields that were available in the three raw data files that were used

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **schoolInfo.json** |  | **salaries-by-region.csv** |  | **states.csv** |
| Ranking |  | School Name |  | State |
| Acceptance-Rate |  | Region |  | Abbreviation |
| Act-Avg |  | Starting Median Salary |  |  |
| Sat-Avg |  | Mid-Career Median Salary |  |  |
| Photo |  | Mid-Career 10th Percentile Salary |  |  |
| Cost after Financial Aid |  | Mid-Career 25th Percentile Salary |  |  |
| City |  | Mid-Career 75th Percentile Salary |  |  |
| Sortname |  | Mid-Career 90th Percentile Salary |  |  |
| Zip |  |  |  |  |
| Percent Receiving Aid |  |  |  |  |
| State |  |  |  |  |
| Average High School GPA |  |  |  |  |
| Tied Ranking |  |  |  |  |
| Public/Private University |  |  |  |  |
| Business Reputation Score |  |  |  |  |
| Tuition |  |  |  |  |
| Engineering Reputation Score |  |  |  |  |
| Enrollment Size |  |  |  |  |
| Region |  |  |  |  |

# Data TransFormation

## Import Data files

Jupyter notebook and Pandas were used to import the two csv files and one json file. The Pandas **pd.read\_csv** and **pd.read\_json** methods were used to import the source data files and convert them into pandas dataframes for further processing.

## Data Cleaning

The data cleaning activities were all done using pandas and includes some of the more challenging and tedious aspects of this ETL project. The steps and some of the challenges are mentioned below

1. **Standardize college names**

The biggest challenge with the source data is that the files are from two different sources (WSJ and USNews) and the names of the universities were not standardized.

As an example, Indiana university is called ***Indiana University (IU), Bloomington*** in one file and ***Indiana University--Bloomington*** in another. To standardize the university names the following operations needs to be done:

* + 1. Remove the parentheses and any text included within
    2. Change ‘-’ to ‘--’
    3. Change ‘,’ to ‘--’

1. **Drop columns** that are not required for the final database
2. **Add columns**

The following id variables are created for use as primary keys in SQL tables: state\_id, college\_id and rankings\_id

1. **Rename existing columns:**

Remove spaces in column names to prepare to be used in the SQL database

## Type conversion

Another challenge is that the various salary fields in the salary file were in string format and has to be converted into float. This way, the user would be able to query the information more easily or perform functions on it. The following steps helped accomplish this.

* + 1. Splice the data in the salary fields
    2. Drop the ‘$’ and ‘,’ from the amounts
    3. Create a float from the resulting text

# Loading into database

## Type of Database

MYSQL database is selected since an open-source relational database was preferred.

## ORM (SQLAlchemy)

SQLAlchemy is used as the ORM for Python to communicate with MYSQL database. Once the SQLAlchemy connection is established, the MYSQL tables can be populated using the ‘to\_sql’ method and read using the ‘read\_sql\_query’ method. Examples are provided below.

Read from MySQL db table in Python:

*state\_list = pd.read\_sql\_query("Select \* from state", con=engine)*

Write to MySQL db table from Python:

*merged.to\_sql(name="college\_list", con=engine, if\_exists='append', index=False)*

## Final DB schema including EER Diagram

Four tables (**rankings college\_list, salary\_region and state**) were created in the final database called **etl\_project.** The fields in each of the final tables and the relationship between the tables are best shown by the EER diagram below. The EER diagram was created in MySQL WorkBench and provides a visual representation of the schema of the database.

