ETL Project
Data Related Job Market

Data Analytics Boot Camp

School of Continuing Studies

University of Toronto

Group Members:

May Ang, Kelvin Deng, Thao Hoang, Yijing Su

Date: December 19th, 2020

TABLE OF CONTENT

LIST OF FIGURES

1. INTRODUCTION

With the rapid advancement of technologies, data related jobs have become highly demanded in recent years. The computer technology boom in the past 20 years has made computer usage much easier and cheaper for everyone, and this has led to the enormous increase in data generation and the demand for data organization. With an easier access to data, and the importance of data in decision making, data analysis has become significantly crucial in all industries. According to the LinkedIn Workforce Report in the USA, the demand for data related professions has multiplied six times compared to five years ago, and this growth will continue for the next five years [1]. Therefore, people with data organization and analysis skills are highly in demand in the work force, and data related topics are currently the main innovation focus in many IT professionals.

This project focuses on the current job market for data related jobs in four different countries: The United States (US), Canada, Australia, and Singapore. Using the Extraction, Loading, Transformation (ETL) methodology, the project will extract all data related job postings from Indeed in these four countries, transform them, and load them into an SQL database. The project scope will include data analyst, data scientist, data engineer, and machine learning roles. Furthermore, this project also used the same ETL process to obtain the world's university ranking data and mental health survey for further analysis on any possible relationships.

2. METHODS

ETL refers to the extraction, transformation, and loading of data. It is the general procedure for data integration, which is the first step of data analysis. ETL is the strategy for extracting data from different sources and organizing them into new datasets for more ease in analysis. Extraction is the first step and it is where data is extracted from different sources. Transformation is the next step. To transform data, the process includes cleaning, summarization, selection, joining, filtering, and aggregating. This ensures that all the data will have a similar format that can be easily analyzed. Lastly, loading is to load the transformed data into a database or into a common folder, where the data will now be ready for data visualization and data analysis.

2.1 Extraction

For this project, data on the current job market for data related jobs in the US, Canada, Australia, and Singapore were required. Specifically, data on current job postings for data analysts, data scientists, data engineers, and machine learning roles. Initially, several csv files related to this project were obtained from websites such as Kaggle. However, upon more investigation, they lacked key information and the data from different countries were not easily comparable. Due to the inconsistency and noisiness of these csv data, it was decided that web scraping from Indeed would be a better option.

The web scrape was performed using four separate Jupyter notebooks located in the Extraction Subfiles folder. Each file is specific to each of the four countries, and within the file, a scrape for each of the job role was done. The web scraping process was done using the BeautifulSoup and Requests module. An API was requested using the Request module, then the BeautifulSoup module was used to retrieve the data stored inside the webpage's HTML. It was also discovered that by using the lxml parser instead of the html.parser, data for more job postings were obtained.

Each job post was located inside a tag under the class: "result". With that, a for loop was used to iterate through every tag under this class to obtain data on each posting's job title, job id, company name and job location. The job title was located within an <a> tag and "jobtitle" class. The job id was located within a tag and "company" class. The job id was located in different tags but all within the same "id" class. The location was

also located in different tags but all within the same “location” class. To obtain only the text inside the code, “.text” or “.text.strip()” was added into the result.find code. The data collected was then stored into its corresponding list.

This for loop was nested inside another for loop that iterated through different pages. Due to high quantities of job postings in some of the requests, a limit of 70 pages (equaling to around 1000 job posts) was set.

Figure 1 shows a snippet of the code.

```
# Scraping to page limit of 70 pages
page = range(0,710,10)

# Changing integer to string
page_string = map(str, page)

for page in list(page_string):
    url = f"https://{country}.indeed.com/jobs?q=Data+Analyst&radius=25&start={page}"
    print(url)
    time.sleep(5)

    # Retrieve page with the requests module
    response = requests.get(url)

    # Create BeautifulSoup object; parse with 'lxml'
    soup = BeautifulSoup(response.text, 'lxml')

    # Retrieve the parent divs for all postings
    results = soup.find_all('div', class_='result')

    # loop over results to get job posting data
    for result in results:
        try:
            job_title = result.find('a', class_='jobtitle').text.strip()
            job_index = 1
            company = result.find('span', class_='company').text.strip()
            job_id = result.get('id')
            location = result.find(class_='location').text

            # Adding data to lists
            job_title_list.append(job_title)
            job_title_index.append(job_index)
            company_list.append(company)
            location_list.append(location)
            job_id_list.append(job_id)

        except:
            pass
```

Figure 1. Web scraping Code Snippet

A challenge imposed during the extraction step was the banning from Indeed during web scraping. Indeed recognized the ongoing data extraction and banned the web request due to a block from the robot detection program. This was likely triggered by the excessive number of requests. To solve this, separate files were created for each country to run at separate times as well as a time.sleep function was added within the for loop to slow the computer down. By slowing down the requests and randomizing how long the sleep is for,

data was successfully obtained from the Canadian, US, and Singaporean website. However, the Australian website had to be resolved using another method.

For the Australian data extraction, the user agent was replaced for every 10 web requests (i.e. avoid using the same browser identifier throughout the scraping process). By doing so, the website will think that different users are searching for each job instead of a single user searching for all roles. This was successful in preventing the ban.

A library called "fake_useragent" was installed to generated random user agent. A random user agent example is listed below:

```
1. Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_0) AppleWebKit/537.36
   (KHTML, like Gecko) Chrome/32.0.1664.3 Safari/537.36
```

To better disguise the web scraping requests, a couple of methods were also utilized to fake random IP addresses to avoid being banned for hours or days. From a free proxy website (https://www.sslproxies.org/), a list of free HTTPS proxies was web scraped for the use of web requests through proxies.

However, most of the HTTPS proxies have slow connections, which affected the efficiency of web scraping. A VPN application on Windows platform called Tunnel Bear was used instead for a faster connection. And results were excellent without being banned by Indeed anymore.

For the Mental Health Survey data and University Ranking data, a simple Pandas function was used to read the csv in and into a dataframe. Figure 2 shows a snippet of the code.

```
university_filepath = "../Raw Data/UniversityData.csv"
mentalhealth_filepath = "../Raw Data/MentalHealthSurvey.csv"

uni_data = pd.read_csv(university_filepath)
uni_df = pd.DataFrame(uni_data)

mh_data = pd.read_csv(mentalhealth_filepath)
mh_df = pd.DataFrame(mh_data)
```

Figure 2. Extraction of Mental Health Survey data and University Ranking data.

2.2 Transformation

Transformation is an important step of ETL as it is the step that consolidates all the data together. It uses methods such as data cleaning to change all the data into the same format

and data indexing to optimize data storage and computer RAM. How data is transformed depends on the requirements of the project as well as what kind of analysis will be done with the data.

For this project, the goal was to transform the data from multiple sources into the same format and prepare them into tables with foreign key connections for loading into a SQL database. Since this ETL process involves the process of converting a NoSQL data into a SQL database, data normalization will be necessary. An entity relationship diagram (ERD) was created to visualize the relationships and table structures and is explained in more detail in Section: 2.3 Loading.

In the job market data (from web scraping), data cleaning was done to remove duplicates. Since the job ID was unique to each posting, it was used to remove the duplicates. Since a job ID could have shown up in several searches (e.g. one job showed up in both data analyst and data engineer searches), a hierarchy was established based on how specific the job role was. The hierarchy is as follows: Machine Learning, Data Engineer, Data Scientist, and Data Analyst. For example, if a job ID was labelled as both machine learning and data analyst, the entry for data analyst will be removed.

The location in the data required cleaning up as well. Since many of the location data included both city and state, it was further split into two separate “City” and “State” columns. This was done using a string splitter. Once cleaned up, the four transformed job market data (for each country) were combined into one data frame.

For the university ranking (institution) data and mental health data, data cleaning was performed to remove any unrelated data such as information on other countries besides the four and past data (keeping only the most recent information). There were also many blank values that were with “N/A”.

For data normalization, certain columns were cleaned up into more consistent responses. For example, in the mental health survey data, the gender column included various spellings for “Female”. During transformation, the responses were cleaned up into only three unique values of “Female”, “Male”, and “Other”. This is shown in the code in Figure 3.

```
#normalizing gender entry
female = ["Female", "female", "F", "f", "Female ", "Femake", "Trans woman", "Cis Female", "Trans-female", "cis-female/femme",
          "queer/she/they", "Trans-female" "Cis Female", "Woman", "woman", "Female (trans)", "Female (cis)", "femail"]
male = ["M", "Male", "male", "m", "Male-ish", "maile", "Cis Male", "Male (CIS)", "Make", "male leaning androgynous", "Male ",
        "Man", "Mail", "msle", "cis male"]
other = ["Guy (-ish) ^_^", "p", "non-binary", "Nah", "Genderqueer", "Other"]

for gender in female:
    #replacing with country index number
    df["Gender"]= np.where((df.Gender == gender), "Female", df.Gender)

for gender in male:
    #replacing with country index number
    df["Gender"]= np.where((df.Gender == gender), "Male", df.Gender)

for gender in other:
    #replacing with country index number
    df["Gender"]= np.where((df.Gender == gender), "Other", df.Gender)

df["Gender"].unique()
```

Figure 3. Column Normalization

Besides the data transformation on the extracted data, index tables to show the relationship between the datasets. The country table and the job title were created and the proper index numbers replaced the appropriate values in the job market data as well as the institution and mental health datasets.

job_title_id	job_title
1	Data Analyst
2	Data Scientist
3	Data Engineer
4	Machine Learning

country_id	country_name
1	Singapore
2	Canada
3	United States
4	Australia

Figure 4. Index Tables for Job Title and Country.

With the separate location summary table created from splitting the company location (city and state columns) information, a unique location ID could be created. Duplicates were dropped to ensure uniqueness and its corresponding latitude and longitude values were obtained using the API, api.opencagedata.com. This code is in the city_locaton_API in the transformation subfile. As the consolidated location summary table was needed, the API pull was done during transformation. This index was used to replace the “city” and “state” columns in the main job market dataset.

2.3 Loading

Loading is the final step of the ETL process, and it is to load the transformed data into a SQL database. After the data transformation, a complete relationship between each transformed table and intermediate table was established. SQLAlchemy in conjunction with Pandas was used to import data to the databases. PostgresSQL and ElephantSQL were chosen because of its flexibility to allow users to control table structures, datatypes and relationships between each table within a certain database.

To have a direct and clear visualization of the relationship between each table, an ERD was created and is shown below.

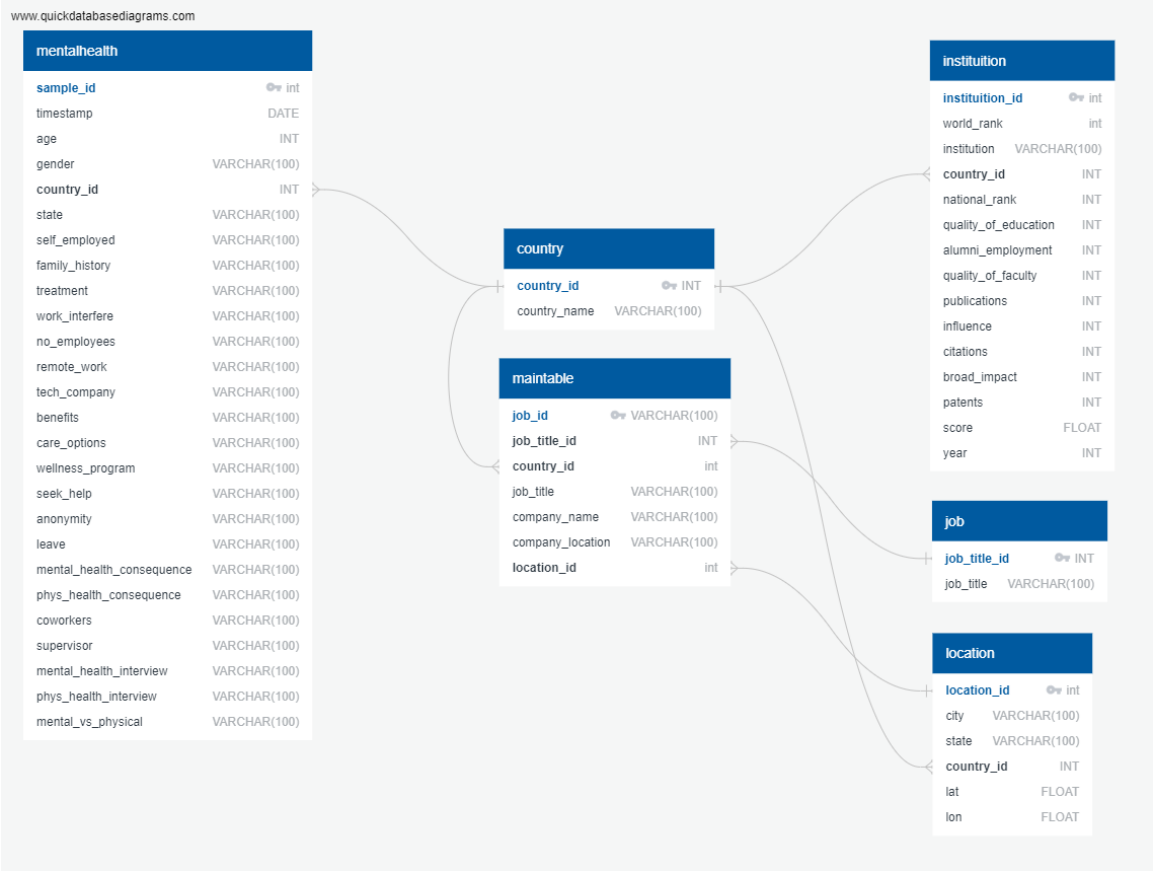

Figure 5: ERD

Based on the relationship displayed in the ERD, relational tables were created and imported directly using SQLAlchemy on Python. An example is shown below:

```
class Main(Base):
    __tablename__ = 'maintable'
#     __table_args__ = {'extend_existing':True}

    job_id = Column(String, primary_key=True)
    job_title_id = Column(Integer, ForeignKey("job.job_title_id"))
    country_id = Column (Integer, ForeignKey("country.country_id"))
    job_title = Column(String)
    company_name = Column(String)
    company_location = Column(String)
    city = Column(String)
    state = Column(String)
```

Figure 6, Relational Tables

To accomplish this task, all csv files obtained from transformation step were loaded and transformed into data frames using Pandas. An example is shown below:

```
main_table_df = pd.read_csv("Transformed Data/AllJobMarket-Transformed.csv")
main_table_df.head()
```

Figure 7. CSV to Data Frame for Loading

An engine was created to establish a route to allow access and query of database by using Python. With this engine created, classes could be instantiated by using the declarative base class to create relational tables in the SQL database using Metadata.

Sessions are created to make connections between the program to the database which enables pandas.to_sql() to commit dataframes into the database. Postgres SQL username and password were stored into the file named config.py. This file was not pushed to the Github to hide the confidential information from the public users and for users running this code, they will require the set up their own config.py file.

Even though SQLAlchemy is a powerful module to interact with a database, users need to have a good understanding of what each step does and how they need to be performed to make it work. The following figure helped us to visualize the concept better.

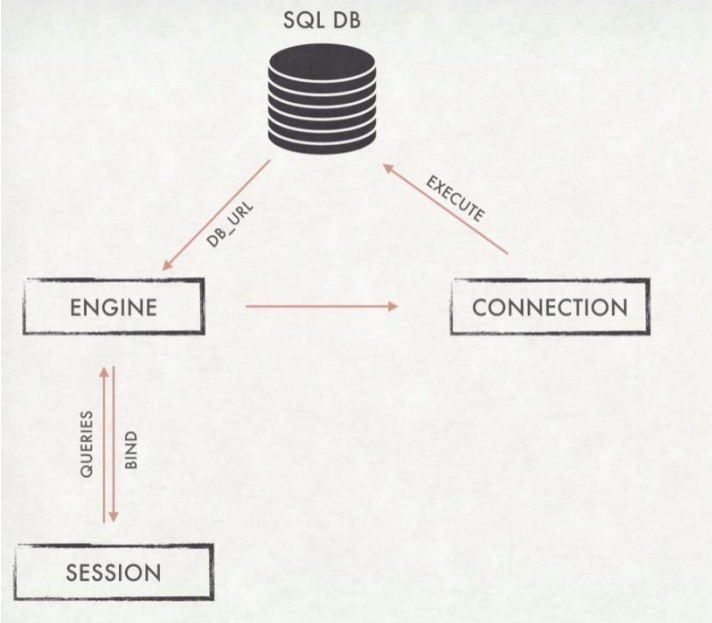

Figure 8. Database Connection

Since users are using different computers to access the database, a locally hosted database would not be accessible. To resolve this, Ngrok was used. Ngrok can expose local servers to the internet, allowing users to access the same database with their own computer.

For users to access the database, users must create a server connection on pgAdmin using the following host address and port.

```
ngrok by @inconshreveable

Session Status                online
Account                       Kelvin Deng (Plan: Free)
Version                       2.3.35
Region                        United States (us)
Web Interface                 http://127.0.0.1:4040
Forwarding                    tcp://2.tcp.ngrok.io:16217 -> localhost:5432

Connections                   ttl     opn     rt1     rt5     p50     p90
                              2       2       0.00    0.00    1.89    3.33
```

Figure 9. Ngrok dashboard information

The main drawback of Ngrok host is that the host computer must be online, as a result, the ElephantSQL cloud database was also used to store the project data. Cloud database such as ElephantSQL has the benefits of storing data on the cloud and allow the ease of access by different users through internet. It can also allow access while the host is offline and

through a web interface or vendor's API, is easily scalable, and most importantly provide a security of the data through backups on remote servers.

In the load.ipynb, two options of database connections were given to load the data as shown below.

```
# Create engine as an interface to PostgreSQL database named Project_2_
ETL
db_local = f'postgresql://{username}:{password}@localhost:5432/Project_
2_ETL'

# An ElephantSQL link was also imported for cloud database access
db_link = "postgres://irxrnnfx:8E4uvAlptBYdblhx20hS5t_elOsRbOxm@suleima
n.db.elephantsql.com:5432/irxrnnfx"
db = create_engine(db_link)
```

Figure 10 shows the stats of the ElephantSQL database, including all six tables with all rows.

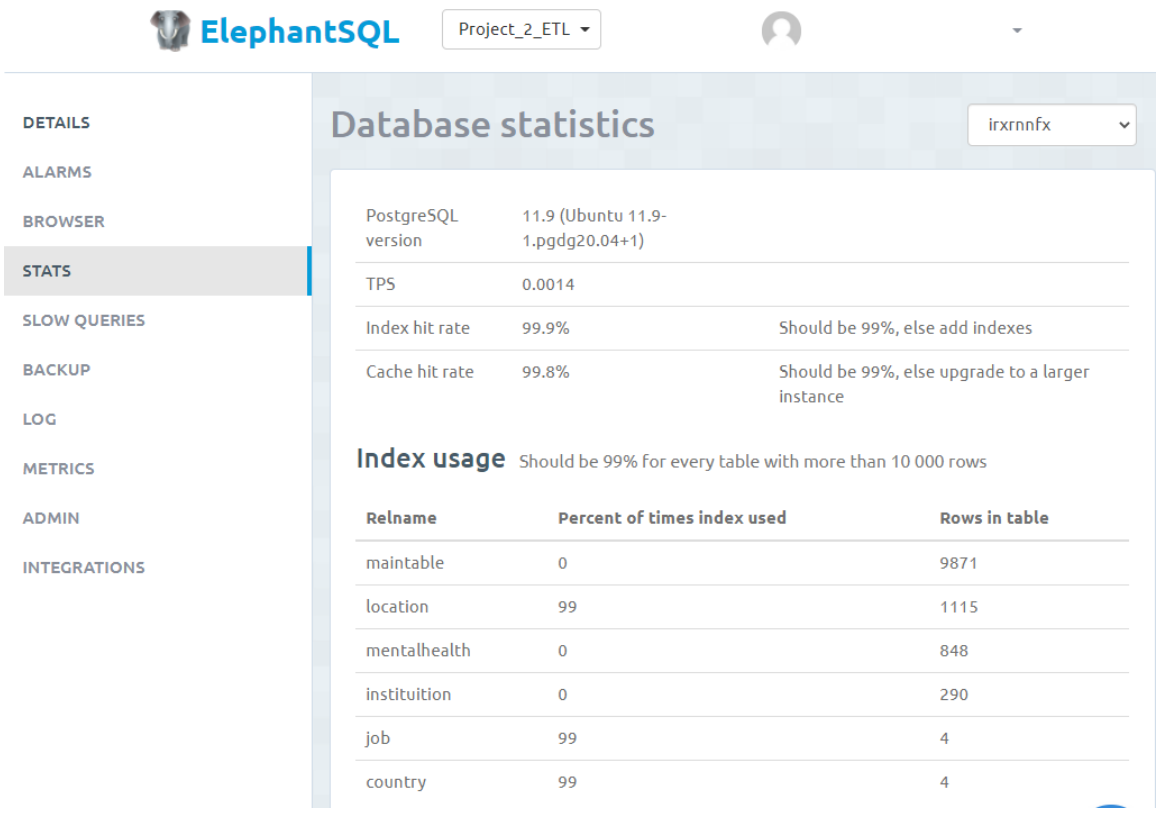

Figure 10. ElephantSQL database stats

Figure 11 shows an example of SQL query command in the ElephantSQL database.

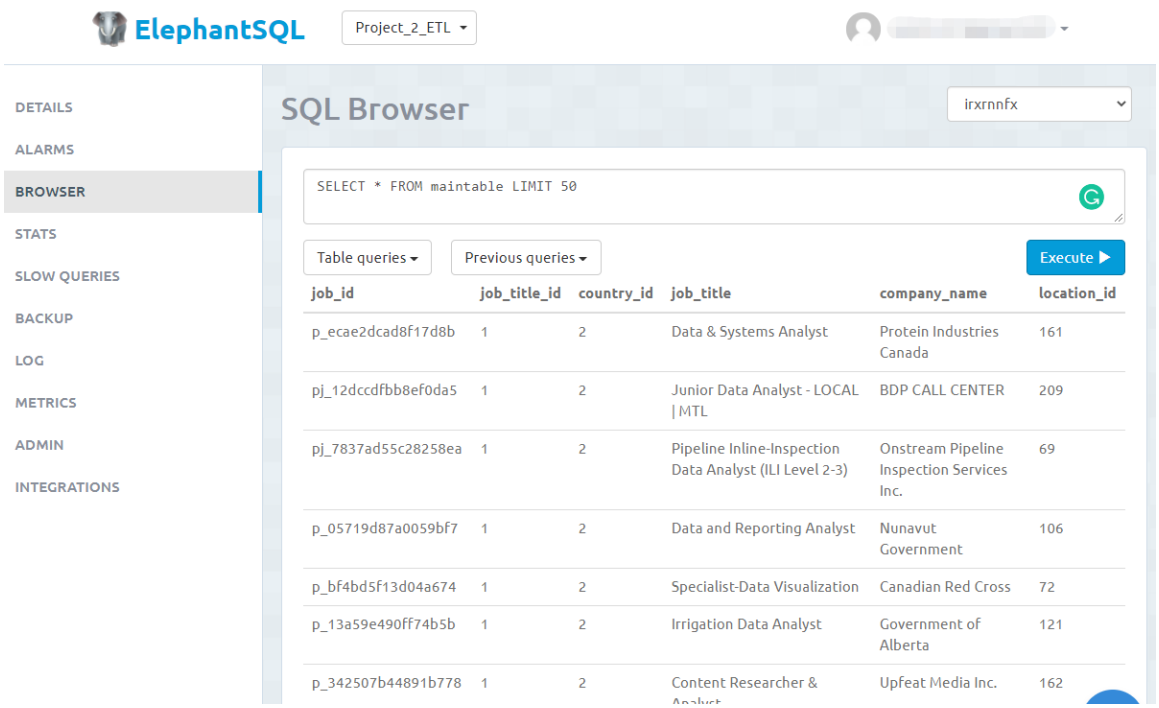

Figure 11.SQL query in ElephantSQL

3. CONCLUSION

This project scope focused on the current job market data for data analysists, data scientists, data engineers and machine learning in four different countries: US, Canada, Australia, and Singapore. The job market data of each country was successfully extracted by web scraping from the Indeed website. The world's university ranking data and the mental health survey were also extracted successfully from online as csv files. Data transformation and normalization were performed on all the data and several intermediate index tables were created to complete key relationships between each data. Lastly, all the transformed data and intermediate tables were loaded into a PostgresSQL database based on their relational order and as seen in the ERD. By performing this ETL process to convert all topic related data extracted from different sources into one SQL database with direct relationships, the data querying and analysis in the next steps of this analysis will take place smoother and with more ease on the computer RAM.

It is also important to note that web scraping at a different time will yield different job postings. Thus, our project is limited to the jobs posted on December 15, 2020.

A main.ipynb file has been created to run all notebooks.

4. FUTURE WORK

Based on the time constraint and the lack of knowledge in higher level of data analysis, there are some limitations and improvements that can be considered to improve the accuracy and completeness of the data.

Another improvement is to move the database onto a cloud-based database. Currently, all the data is stored on a locally hosted PostgresSQL database that is accessible through Ngrok to other users, while the host is online.

Included in the next steps will be the actual analysis of the data. The project objective is to understand the job market in the four countries and also to look into other relationships like how universities in those countries rank, as well as the mental health of employees in technology companies in those countries. Thus, the next step will be using those relationships within the different tables to investigate if there are any correlations.

REFERENCE

[1] Newsroom, "Data Analyst, the most in-demand job of the coming years," 2018 February 21. [Online]. Available: https://www.morningfuture.com/en/article/2018/02/21/data-analyst-data-scientist-big-data-work/235/.

[2] Purdue University Global, "Rise of the Data Analyst—What's Behind the Boom?," 16 September 2019. [Online]. Available: https://www.purdueglobal.edu/blog/information-technology/rise-of-data-analyst/.

[3] edureka, "10 Reasons Why Big Data Analytics is the Best Career Move," 23 July 2020. [Online]. Available: https://www.edureka.co/blog/10-reasons-why-big-data-analytics-is-the-best-career-move.

[4] SAS, "ETL What it is and why it matters," [Online]. Available: https://www.sas.com/en_ca/insights/data-management/what-is-etl.html.

[5] Open Sourcing Mental Illness, LTD, "Mental Health in Tech Survey," 2014. [Online]. Available: https://www.kaggle.com/osmi/mental-health-in-tech-survey. [Accessed 15 December 2020].

[6] M. O'Neill, "World University Rankings," 2019. [Online]. Available: https://www.kaggle.com/mylesoneill/world-university-rankings. [Accessed 15 December 2020].

[7] Indeed Singapore, "Indeed Singapore," [Online]. Available: https://sg.indeed.com/. [Accessed 15 December 2020].

[8] Indeed Canada, "Indeed Canada," [Online]. Available: https://ca.indeed.com/. [Accessed 15 December 2020].

[9] Indeed Australia, "Indeed Australia," [Online]. Available: https://au.indeed.com/. [Accessed 15 December 2020].

[10] Indeed US, "Indeed US," [Online]. Available: https://www.indeed.com/. [Accessed 15 December 2020].