# Purpose

The purpose of this program is to allow a user to determine what are the most common keywords in job descriptions for specific job titles which they are seeking. It does this by searching job descriptions on indeed based on user input keywords, then creates a count of words and phrases (up to four words) used in job descriptions .

# Storing the Data

In this project, I created a Postgres Database on Amazon Web Services RDS (Relational Database Service). The advantage of this platform is the storage space and computational space is scalable as needs increase.

Postgres has a well maintained interface library – Psycopg2 which allows us to work easily with it in Python.

# Searching Indeed

## Process Overview

In the overall process the flow is as follows

1. The *indeed\_search\_executer.py* is the first program to start.
   1. *indeed\_search\_executer.py*  queries **indeed\_search\_set** to return a result set where the field *search\_completed* is False.
   2. For each record in the result set, *indeed\_search\_function.py*  runs on that element
   3. *indeed\_search\_function.py*  returns the row information modified to show if the search was successfully completed or not.
   4. The data is committed and closed.

## Traversing and executing a list of searches to be done.

*indeed\_search\_executer.py* goes through the database table **indeed\_search\_set**. This table holds the two critical elements for an indeed.com search, the search keyword list and the search zip code. It also has a seach\_completed flag which is flipped when the search is successfully completed. The table **indeed\_search\_set is** defined in the remulak database as

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Comments** |
| creation\_date | date | Timestamp of the creation of the search date |
| Iss\_pk | Serial | Primary Key for the table |
| search\_completed | Boolean | A Boolean flag flipped to true when a search is completed |
| search\_keyword\_list | Varchar(63) | List of keywords to search. These will be concatenated. |
| Search\_run\_date | Date | The actual date the search is run. |
| search\_zip\_code | Varchar(31) | At now, this will only support a single zip code. Future features should be able to take a list of zip codes or geographic location. The only problem with geographic location is the comma in city state which must be converted to a %2C character for inclusion in to the python string. |

This program uses [SQLAlchemy](https://www.sqlalchemy.org/) to work with Postgres tables as objects. Therefore we define classes which mirror each of the tables in the remulak database. Below is the class “Serch”[sic.] which mirrors the database table ‘**indeed\_search\_set’**

class Serch(Base):  
 \_\_tablename\_\_ = 'indeed\_search\_set'  
 creation\_date = Column(Date)  
 iss\_pk = Column(Integer, primary\_key=True)  
 search\_completed = Column(Boolean)  
 search\_keyword\_list = Column(String)  
 search\_run\_date = Column(DateTime)  
 search\_zip\_code = Column(String)

## Building a table of search results.

Building a table of search results is done by the function indeed\_search\_function.py. Searching indeed is a two-step process. First a list of results for a given combination is found, then the URLs from those search results are opened later and the job description text parsed.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Data Type** | **Comments** |
| company | VARCHAR(64) | The extracted company. |
| extracted\_url | Object | The URL with only the essential information to retrieve the job found. No external tracking |
| guid | VARCHAR(32) | The guid is a unique id assigned to a job by indeed |
| isr\_pk | Serial(int) | The Primary Key for the results |
| iss\_pk | Integer | This is the foreign key to table indeed search set so we know which search created these results. |
| Job\_title\_row | Text | All of the information in the title line of indeed |
| lat | Float64 | Latitude from the listing |
| longitude | Float64 | Longitude from the listing |
| publish\_date | Timestamp | The derived timestamp from which it was published. |
| scraped | Bool | The flag which is flipped to True when a job is scraped |

The class Rezult is the corresponding SQLAlchemy class for **indeed\_search\_results.**

class Rezult(Base):  
 \_\_tablename\_\_ = 'indeed\_search\_results'  
 company = Column(String)  
 extracted\_url = Column(String)  
 guid = Column(String)  
 isr\_pk = Column(Integer, primary\_key=True)  
 iss\_pk = Column(Integer)  
 job\_title\_row = Column(String)  
 latitude = Column(Float)  
 longitude = Column(Float)  
 publish\_date = Column(DateTime)  
 scraped = Column(Boolean)

Indeed\_search\_function.py is passed an object of the class Rezult. Postgres Database and Web sessions are created. A string which is created from the keyword and zip code and a starting point is created and sent to the session. The contents are returned and parsed by BeautifulSoup with the html5lib.

Search results from indeed.com come in an RSS XMLformat making parsing easier. Each job is listed as an ‘item’ that item is what is searched for. The result set is iterated through and mapped to the appropriate variables with a Rezult class method, added to the queue and flushed to the Database.

After the loop is over, the information is committed and quit.

Upon searching the results it was found there were a lot of duplicates. An explicit duplicate removal SQL string is sent directly to the database.

# Appendix A

Required modules in Alphabetical order

|  |  |  |
| --- | --- | --- |
| Module | Location | Version used |
| Pandas | <https://pandas.pydata.org/>,<https://pypi.org/project/pandas/> | 0.25.1 |
| pathlib | Standard Module |  |
| getpass | Standard Module |  |
| objcrypt | <https://pypi.org/project/objcrypt/> | 0.6 |
| SQLAlchemy | <https://pypi.org/project/SQLAlchemy/> | 1.3.8 |