Mindex Data & Analytics Code Challenge

Background:

In 2021, the Cincinnati Bengals came so close to winning their first NFL Championship (make sure to ask John Hill about it if you come visit our office)! In this challenge, you will be asked to analyze the team's data from that year. We will provide you access to 4 datafiles:

- *bengals.csv is a catalog of the Bengals wins and losses.*
- *boyd_receiving.csv is a catalog of pass catcher Tyler Boyd's stats.*
- *chase_receiving.csv is a catalog of pass catcher Ja'Marr Chase's stats.*
- *higgins_receiving.csv is a catalog of pass catcher Tee Higgins's stats.*

You will be tasked with retrieving these files from an AWS environment, performing transformations and joins on the data, and storing them in a database. You will then write a query and execute it against your database. This challenge is meant to mimic an Extract, Transform, Load (ETL) project that you may encounter at Mindex. Your code should be version controlled and committed frequently to a public Github repo.

Requirements:

- pip installed on computer
- jupyter notebook installed ('pip install notebook')
- pandas installed ('pip install pandas')
- boto3 installed ('pip install boto3')

Instructions:

1. Create a new Jupyter Notebook project and store it in a public Github repo. Throughout this exercise, make sure to frequently commit your work!
2. Use the boto3 library to configure access to a Mindex AWS S3 bucket
    - Credentials - access key ID: AKIAZZ33YB65GZIN656A
      secret access key: i4RvJxZXAw1pOFMRdKp3Jp2c3x+BHiGfVEWi+ZKA
3. Use given credentials to download the following four CSV files from the 'mindex-data-analytics-code-challenge' S3 bucket. This data represents the results of the Bengals 2021 NFL Season and the stats of the team's top 3 receivers.
    - bengals.csv
    - boyd_receiving.csv
    - chase_receiving.csv
    - higgins_receiving.csv

4. Use the pandas library to load each CSV into its own dataframe.
5. Join/Merge all of the dataframes together to display one global table that shows the three different receiver's yards and touchdown (TD) data as well as every game result. Be sure to include *Opponen*t, *Location*, and *Result* fields from the bengals.csv file.
    - Make any changes necessary to successfully merge/join the dataframes.
6. Replace the '1.0' or '0.0' values in the Result field to display 'Win' or 'Loss', respectively.
7. Write the dataframe to a postgresql DB (credentials below). A username, password, and table name have already been created with your name. There are several options you can use: SQLAlchemy, psycopg2, Pandas. We encourage you to use whichever you find most suitable just note that you may need to install additional Python packages.
    - Database host: ls-2619b6b15c9bdc80a23f6afb7eee54cf0247da21.ca3yee6xneaj.us-east-1.rds.amazonaws.com
    - Username: 'your first name'_'your last name'
        i. Example: john_hill
    - Password: your first letter of your first name swapped with the first letter of your last name
        i. Example: John Hill would be hohnjill
    - Table Name you are writing to: 'your first name'_'your last name'
        i. Example: john_hill
    - Database name: 'postgres'
8. Set up a connection to view the database using DBeaver and ensure all relevant data is present. Write a SQL Query to show the total yards each receiver had throughout the season as well as the team's record displayed as: "# of Wins - # of Losses". Your query should generate the following view:

Boyd Yards	Higgins Yards	Chase Yards	Win/Loss
938	1,400	1,823	14-10

9. Make one final commit of your Jupyter Notebook and your SQL Code to your public Github Repo and share the link with us!