

Day60 - April 5th 2024

1. Started my day as usual
2. Solved one leetcode medium problem
3. Started implemented data engineering project (AWS)

The screenshot shows a YouTube video player on the left and a Google Docs document on the right. The YouTube video is titled "Zillow Data Analytics (RapidAPI) | End-To-End Python ETL Pipeline | Data Engineering Project |Part 1" by tupleps. The video has 15K views and was uploaded 7 months ago. The Google Docs document is titled "4,5-4-24_ETL Pipeline" and contains Python code for an ETL pipeline. The code includes comments and function definitions for extracting data from Zillow's RapidAPI and saving it to a file.

```
# Load JSON config file
with open('/home/ubuntu/airflow/config_api.json', 'r') as config_file:
    api_host_key = json.load(config_file)

extract_zillow_data = PythonDag
task_id = 'task_extract_zillow_data'
python_callable = extract_zillow_data

def extract_zillow_data(**kwargs):
    url = kwargs['url']
    headers = kwargs['headers']
    querystring = kwargs['querystring']
    # string = kwargs['data_string']
    # return headers
    response = requests.get(url, headers=headers, params=querystring)
    response_data = response.json()

    # Specify the output file path
    output_file_path = f'/home/ubuntu/response_data_{kwargs['data_string'].json}'
    file_str = f'response_data_{kwargs['data_string'].json}'

    # Write the JSON response to a file
    with open(output_file_path, 'w') as output_file:
        json.dump(response_data, output_file, indent=4) # indent for pretty formatting
    output_list = [output_file_path, file_str]
    return output_list

10. Next we will concentrate on python callable function

11. Here in our code we use kwargs are

12. We have used data_string ...it is used to give naming for our extracted data...see and code

14. Now lets test our DAG
```

4. Please find the details of the doc here : [4,5-4-24_ETL Pipeline](#)

5. Ended my day by solving SQL Question from Hackerrank

HackerRank | Prepare | SQL | Advanced Join | Placements

Exit Full Screen View

Problem

ID and Name: Friends contains two columns: ID and Friend_ID (ID of the Other Best friend).
Packages contains two columns: ID and Salary (offered salary in \$ thousands per month).

Column	Type
ID	Integer
Name	String

Students

Column	Type
ID	Integer
Friend_ID	Integer

Friends

Column	Type
ID	Integer
Salary	Float

Packages

Write a query to output the names of those students whose best friends got offered a higher salary than them. Names must be ordered by the salary amount offered to the best friends. It is guaranteed that no two students got same salary offer.

Sample Input

AI Assistant is here to help!

```
1 /*
2  Enter your query here.
3  */
4
5  select s.name from Students S
6  inner join Packages P on S.ID = P.ID
7  inner join Friends F on S.ID = F.ID
8  inner join Packages P1 on F.Friend_ID = P1.ID
9  where P1.salary > P.salary
10 order by P1.salary
11
12 /*
13  inner join Friends F on S.id = F.ID
14  inner join Packages P1 on F.ID = P1.ID*/
15
16
```

Line: 10 Col: 19

Upload Code as File

Run Code Submit Code

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Your Output (stdout)

```
1 Stuart
2 Priyanka
3 Paige
4 Jane
```