AWS Lambda

Note: Please replace **<your bucket name>** below to your own bucket name!

1. Create a lambda function with the code below (name it as **remove_feature_files** and choose **Python 3.8** as runtime environment), make sure you assigned **AmazonS3FullAccess** and **AmazonAthenaFullAccess** to the function role and increase the function timeout to **1 minute**.

```
import json
import boto3
def lambda_handler(event, context):
    # TODO implement
    bucket = event['bucket']
    prefix = event['prefix']
    s3 = boto3.resource('s3')
    bucket = s3.Bucket(bucket)
    for key in bucket.objects.filter(Prefix=prefix):
        key.delete()
    return {
        'statusCode': 200
    }
```

Execute this function by clicking the test button on the top right of the page, for the test events please put below json object:

```
{
        "bucket": "<your bucket name>",
        "prefix": "features/"
}
```

Note: you need to replace <your bucket name> to the name of the bucket you created

2. Create another lambda function called **exe_query_order_products_prior** with the code below, please **re-use the role** you created before and increase the function timeout to **1 minute.** Please examine the query1 and query2 parameter, this function is intended to drop the relevant table if exists and re-create them.

```
import json

import boto3

import time


athena_client = boto3.client('athena')


def lambda_handler(event, context):


    database = event['database']

    query_output = event['query_output']

    # TODO implement

    query1 = """

    DROP TABLE IF EXISTS order_products_prior

    """

    query2 = """

    CREATE TABLE order_products_prior WITH (external_location = 's3:// <your s3 bucket>/features/order_products_prior/', format = 'parquet')

    as (SELECT a.*,

        b.product_id,

        b.add_to_cart_order,

        b.reordered

    FROM   orders a

        JOIN order_products b

          ON a.order_id = b.order_id
```

```
WHERE  a.eval_set = 'prior')
"""
response1 = athena_client.start_query_execution(
    QueryString=query1,
    QueryExecutionContext={
        'Database': database
    },
    ResultConfiguration={
        'OutputLocation': query_output
    }
)

# sleep 10 seconds to make sure the table is successfully dropped
time.sleep(10)

response2 = athena_client.start_query_execution(
    QueryString=query2,
    QueryExecutionContext={
        'Database': database
    },
    ResultConfiguration={
        'OutputLocation': query_output
    }
)

# get the query execution id
execution_id = response2['QueryExecutionId']

while True:
```

```
    stats = athena_client.get_query_execution(QueryExecutionId=execution_id)
    status = stats['QueryExecution']['Status']['State']
    if status in ['SUCCEEDED', 'FAILED', 'CANCELLED']:
        break
    time.sleep(0.2)  # 200ms


return {
    'statusCode': status
}
```

Execute the function and put below json object as input:

```
{
  "database": "prd",
  "query_output": "s3://<your s3 bucket>/query_results/"
}
```

Note: you need to replace <your bucket name> to the name of the bucket you created. You may have noticed in the previous lambda function, query1 is merely to drop the table (order_products_prior) if exists, query2 is to re-create the table.

3. Please create 4 more lambda functions(exe_query_user_features_1, exe_query_user_features_2, exe_query_up_features and exe_query_prd_features) similar to above, but change the table name and query to the ones you did in project_part2.
   (Note: before testing these functions, make sure there are no files in this location 's3:// <your s3 bucket>/features/<your table name> by running the remove_feature_files function, otherwise you cannot re-create the table)