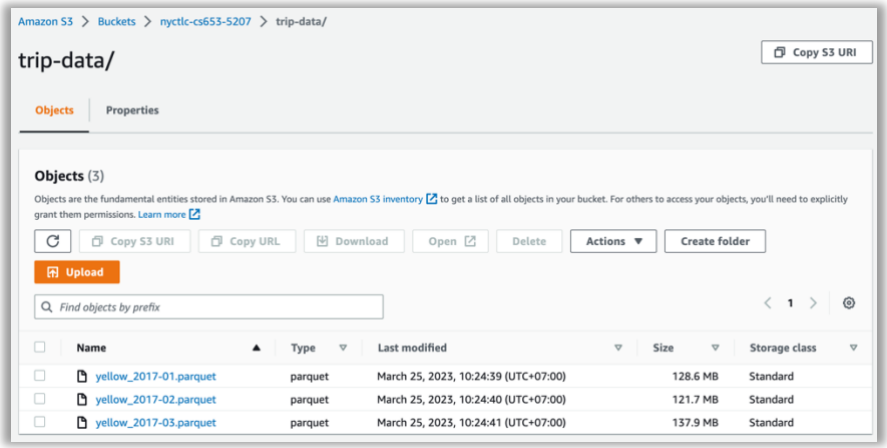
6509035207 KUNANON SUKJARUEN

HW2 Yellow Taxi

- ภาพหน้าจอที่แสดงให้เห็น s3 bucket nyc-tlc-cs653-xxxx ต้องเห็นชื่อ bucket และข้อมูลข้างใน bucket



ขั้นตอนที่ 1 เป็นการ Import boto3 เพื่อนำเข้าไลบรารี boto3 ซึ่งเป็นไลบรารี Python สำหรับทำงานกับบริการของ Amazon Web Services (AWS) และ Amazon S3

```
import boto3

s3 = boto3.client('s3')

resp = s3.select_object_content(
    Bucket='nyctlc-cs653-5207',
    Key='trip-data/yellow_2017-01.parquet',
    ExpressionType='SQL',
    Expression="SELECT payment_type FROM s3object",
    InputSerialization={'Parquet': {}},
    OutputSerialization={'CSV': {}},
)
```

- s3 = boto3.client('s3') คือ การสร้าง Amazon S3 client จาก boto3 สำหรับสื่อสารกับบริการ S3
- resp = s3.select_object_content(...) คือ การเรียกใช้เมธอด select_object_content บน Amazon S3 client เพื่อสืบค้นข้อมูลใน S3 โดยใช้ S3 Select ฟีเจอร์
- Bucket='nyctlc-cs653-5207' คือ การระบุชื่อของ Amazon S3 bucket ที่เก็บข้อมูล
- Key='trip-data/yellow_2017-01.parquet' คือ การระบุคีย์ (หรือเส้นทาง) ของไฟล์ข้อมูลที่ต้องการสืบค้นใน S3 bucket
- ExpressionType='SQL' คือ การระบุว่าเรากำลังใช้ SQL-like expression ในการสืบค้นข้อมูล
- Expression="....." คือ SQL expression ที่ใช้เพื่อเลือกคอลัมน์จากข้อมูลในไฟล์
- InputSerialization={'Parquet': {}} คือ การระบุว่ารูปแบบของไฟล์ข้อมูลที่เข้ามา (input) เป็น Parquet

3. query ข้อมูลด้วย Amazon S3 select เพื่อตอบคำถามต่อไปนี้

(a) ในเดือน Jan 2017 มีจำนวน yellow taxi rides ทั้งหมดเท่าไร แยกจำนวน rides ตามประเภทการจ่ายเงิน (payment)

```
 GNU nano 2.9.8                                    query-texi.py

import boto3

s3 = boto3.client('s3')

resp = s3.select_object_content(
    Bucket='nyctlc-cs653-5207',
    Key='trip-data/yellow_2017-01.parquet',
    ExpressionType='SQL',
    Expression="SELECT payment_type FROM s3object",
    InputSerialization={'Parquet': {}},
    OutputSerialization={'CSV': {}},
)

# Process the response to extract the result of the SELECT operation
payment_type_counts = {}
for event in resp['Payload']:
    if 'Records' in event:
        records = event['Records']['Payload'].decode('utf-8').strip().split('\n')
        for record in records:
            payment_type = int(record)
            payment_type_counts[payment_type] = payment_type_counts.get(payment_type, 0) + 1

# Show the result
print("Number of rides by payment type:", payment_type_counts)
```

ผลลัพธ์ โดยการใช้คำสั่ง python3 ตามด้วยชื่อ คำสั่ง.py ที่ได้เขียนไว้

```
[ec2-user@ip-172-31-83-247 ~]$ python3 query-texi.py
Number of rides by payment type: {2: 3144926, 1: 6506189, 3: 46257, 4: 13447, 5: 1}
```

(b) ในเดือน Jan 2017 Yellow taxi rides ในแต่ละจุดรับผู้โดยสาร (Pickup location) เป็นจำนวน rides มากน้อยเท่าไรและมีค่าโดยสารรวมของ rides และจำนวนผู้โดยสารเฉลี่ยต่อ rides ในแต่ละจุดเท่าไร

```
import boto3

s3 = boto3.client('s3')
bucket = 'nyctlc-cs653-5207'
key = 'trip-data/yellow_2017-01.parquet'

def get_query_result(expression):
    select_results = s3.select_object_content(
        Bucket=bucket,
        Key=key,
        Expression=expression,
        ExpressionType='SQL',
        InputSerialization={'Parquet': {}},
        OutputSerialization={'CSV': {}},
    )

    result = 0.0
    for event in select_results['Payload']:
        if 'Records' in event:
            result = float(event['Records']['Payload'].decode('utf-8'))

    return result

for i in range(1, 266):
    count_query = "SELECT count(PULocationID) FROM s3object WHERE PULocationID = {}".format(i)
    fare_query = "SELECT sum(fare_amount) FROM s3object WHERE PULocationID = {}".format(i)
    avg_passenger_query = "SELECT avg(passenger_count) FROM s3object WHERE PULocationID = {}".format(i)

    count = int(get_query_result(count_query))
    fare_sum = get_query_result(fare_query)
    avg_passenger = get_query_result(avg_passenger_query)

    print("No. of rides in Location {} = {}".format(i, count))
    print("Sum fare amount in Location {} = {:.2f}".format(i, fare_sum))
    print("Average no. of passenger in Location {} = {:.2f}".format(i, avg_passenger))
```

ตัวแปร resp['Payload'] ซึ่งจะเป็น list ของ dictionary และมีการ loop ผ่านตัวแปร event ใน list นั้น จากนั้น จะมีการตรวจสอบว่าRecords มีอยู่ใน event หรือไม่ ถ้ามี จะใช้คำสั่ง records = event['Records']['Payload'].decode('utf-8').strip().split('\n')

ในการแยกข้อมูลใน event['Records']['Payload'] และแปลงให้เป็น list ของ string ใน loop นี้ จะ loop ผ่าน record และใช้ PULocationID = int(record) ในการแปลง string ใน record ให้เป็นตัวเลข และใช้ PULocationID_counts[PULocationID] = PULocationID_counts.get(PULocationID, 0) + 1 ในการเพิ่มจำนวนของ PULocationID ใน dictionary PULocationID_counts

จากนั้น จะใช้ library prettytable ในการสร้าง table และแสดงผลข้อมูล โดยมีการสร้าง table ด้วย PrettyTable(["PULocationID", "Number of Rides"]) โดยกำหนด header คือ "PULocationID" และ "Number of Rides" จากนั้น จะมีการ loop ผ่าน PULocationID, count ใน dictionary PULocationID_counts และใช้ table.add_row([PULocationID, count]) ในการเพิ่มข้อมูลใน table ในท้ายสุด จะใช้ table.sortby = "Number of Rides" และ table.reversesort = True ในการเรียงข้อมูลใน table ตาม "Number of Rides" ในลำดับจากมากไปน้อย

Result :

```
[ec2-user@ip-172-31-83-247 ~]$ python3 query-texi-pickup.py
No. of rides in Location 1 = 696
Sum fare amount in Location 1 = 49950.14
Average no. of passenger in Location 1 = 1.40
No. of rides in Location 2 = 7
Sum fare amount in Location 2 = 281.00
Average no. of passenger in Location 2 = 1.57
No. of rides in Location 3 = 32
Sum fare amount in Location 3 = 524.85
Average no. of passenger in Location 3 = 1.66
```

(c) ในเดือน Jan - Mar 2017 มีจำนวน yellow taxi rides ทั้งหมดเท่าไร แยกจำนวน rides ตามประเภทการจ่ายเงิน (payment)

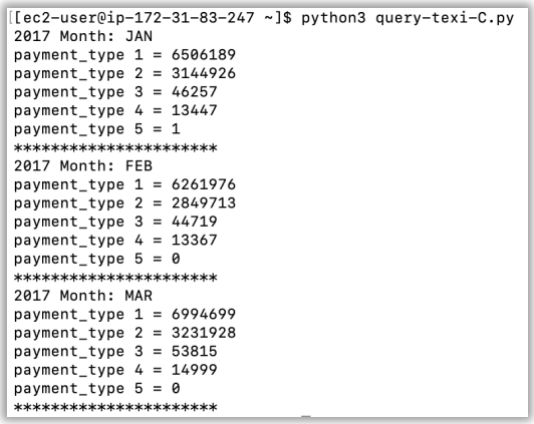
```python
import boto3

s3 = boto3.client('s3')
bucket = 'nyctlc-cs653-5207'

months = ['01', '02', '03']
month_names = ['JAN', 'FEB', 'MAR']

def get_query_result(expression, month):
    select_results = s3.select_object_content(
        Bucket=bucket,
        Key='trip-data/yellow_2017-{}.parquet'.format(month),
        Expression=expression,
        ExpressionType='SQL',
        InputSerialization={'Parquet': {}},
        OutputSerialization={'CSV': {}},
    )

    result = 0.0
    for event in select_results['Payload']:
        if 'Records' in event:
            result = float(event['Records']['Payload'].decode('utf-8'))

    return result

for i, month in enumerate(months):
    print("2017 Month: {}".format(month_names[i]))
    for type in range(1,6):
        query = "SELECT count(payment_type) FROM s3object s WHERE payment_type = {}".format(type)
        count = int(get_query_result(query, month))
        print("payment_type {} = {}".format(type, count))
    print("**********************")
```

Code นี้ แสดงการใช้งานโค้ดในภาษา Python ซึ่งจะทำการ loop ผ่านอาเรย์ months โดยมีจำนวนแถว 3 แถว (01,02,03) และในแต่ละ loop จะใช้ month_names สำหรับแสดงชื่อเดือน (JAN,FEB,MAR) ใน loop นี้ จะใช้ function enumerate() ซึ่งจะทำการนับ index ของ list months และใช้ตัวแปร i และ month ในการ loop โดย month จะเป็นค่าใน list แต่ละรอบ และ i จะเป็น index ของแต่ละรอบ

จากนั้น จะมี loop อีก 1 รอบ ซึ่งจะ loop จาก 1 ถึง 6 และในแต่ละ loop จะใช้คำสั่ง query ในการสร้าง string ซึ่งจะเป็น query SQL สำหรับคำนวณจำนวน payment_type แต่ละประเภท โดยใช้ function format() ในการแทนค่า type ใน query มาจาก query นั้น จะใช้ function get_query_result() ในการเรียก query นั้น โดยส่ง 2 พารามิเตอร์คือ query และ month เพื่อใช้ในการ

คำนวณและค้นหาจำนวน payment_type ของแต่ละเดือน จากนั้น จะมีการเก็บผลลัพธ์จาก get_query_result() ในตัวแปร count และใช้ int() ในการแปลงให้เป็นตัวเลข จากนั้น จะใช้คำสั่ง print() ในการแสดงผลจำนวน payment_type ของแต่ละเดือนและแต่ละประเภท

Result:



การสะท้อนการเรียนรู้ของน.ศ.จากการบ้านครั้งนี้

1. เราได้ความรู้และทักษะอะไรจากการทำการบ้านครั้งนี้บ้าง และคิดว่านำไปใช้ประโยชน์อย่างไรได้บ้าง

จากการบ้านในครั้งนี้ทำให้ได้เรียนรู้เรื่องการนำข้อมูลเข้า และ Query ข้อมูลจำนวนมากมาใช้ โดย Combine ร่วมกับการใช้ภาษา Python ในการได้มาซึ่งข้อมูลที่ต้องการ จากไฟล์ที่มีข้อมูลจำนวนมากๆ ผ่าน Linux

2. สิ่งที่เราชอบและไม่ชอบในการทำการบ้านครั้งนี้

2.1 สิ่งที่ไม่ชอบ คือ การใช้ SQL ใน AWS ผ่านหน้าจอ nano ไม่สามารถใช้บางฟังก์ชันของ SQL ได้ เช่น GROUP BY หรือแม้แต่ ORDER BY ทำให้ต้องใช้ Python ในการได้มาซึ่งคำตอบ ทำให้เกิดความยุ่งยากในการทำงานอย่างยาก

2.2 ไม่มีสิ่งที่ชอบ

3. คิดว่าตัวเองควรปรับปรุงอย่างไร หรือ มีอะไรอย่างอื่นที่ควรได้รับการปรับปรุงสำหรับการบ้านครั้งต่อไป

ต้องฝึกฝนการใช้งาน AWS มากยิ่งขึ้นไปอีก จากคำสั่งและความเข้าใจในการใช้งาน Linux ผ่านTerminal