#AGGREGATED_TRANSACTION

```
import json
import pandas as pd
import glob
import mysql.connector
from sqlalchemy import create_engine
# specify the folder path using '*' pattern
path = 'C:\\Users\\Priyanka\\P!hu\\pulse\\data\\aggregated\\transaction
\\country\\india\\state\\*\\*\\*.json'
files = glob.glob(path)

dfs = []

for file in files:
    folder = file.split("\\")
    data = folder[5]
    type = folder[6]
    country = folder[8]
    state_name = folder[11]
    year = int(folder[12])
    quarter_number = int(folder[13][0])
    quarter = "Q" + str(quarter_number)

    with open(file) as f:
        data = json.load(f)

    dis = []
    for transaction in data['data']['transactionData']:
        name = transaction['name']
        payment_instrument = transaction['paymentInstruments'][0]
        count = int(payment_instrument['count'])
        amount = int(payment_instrument['amount'])
        dis.append({'country': 'India','state': state_name,'year': year, 'quarter':
quarter,'transaction_type': name,'transaction_count': count, 'total_amount': amount})

    df = pd.DataFrame(dis)
    dfs.append(df)

df_agg_trans = pd.concat(dfs)
df_agg_trans = df_agg_trans.reset_index(drop=True)

df_agg_trans.to_csv("C:\\Users\\Priyanka\\P!hu\\agg_trans.csv")
print(df_agg_trans)
```

#AGGREGATED_USER

```
import json
import pandas as pd
import glob
# specify the folder path using '*' pattern
path = 'C:\\Users\\Priyanka\\P!hu\\pulse\\data\\aggregated\\user\\count
ry\\india\\state\\*\\*\\*.json'
files = glob.glob(path)

dfs = []

for file in files:
    folder = file.split("\\")
    country = folder[8]
    state_name = folder[11]
    year = int(folder[12])
    quarter_number = int(folder[13][0])
    quarter = "Q" + str(quarter_number)

    with open(file) as f:
        data = json.load(f)

    dis = []
    for d in data['data']['aggregated']:
        registered = int(data['data']['aggregated']['registeredUsers'])
        app = int(data['data']['aggregated']['appOpens'])

        dis.append({'country': 'India','state': state_name,'year': year, 'quarter': 
quarter,'registered_users':registered,'apps_opened': app})

    df = pd.DataFrame(dis)
    dfs.append(df)

df_agg_user = pd.concat(dfs)
df_agg_user = df_agg_user.reset_index(drop=True)

df_agg_user.to_csv("C:\\Users\\Priyanka\\P!hu\\agg_user.csv")
print(df_agg_user)


#MAP_TRANSACTION

import json
import pandas as pd
import glob
# specify the folder path using '*' pattern
path = 'C:\\Users\\Priyanka\\P!hu\\pulse\\data\\map\\transaction\\hover
\\country\\india\\state\\*\\*\\*.json'
files = glob.glob(path)

dfs = []

for file in files:
    folder = file.split("\\")
    state_name = folder[12]
    year = int(folder[13])
    quarter_number = int(folder[14][0])
    quarter = "Q" + str(quarter_number)

    with open(file) as f:
        data = json.load(f)

    dis = []
    for d in data['data']['hoverDataList']:
        name = d['name']
        metric = d['metric'][0]
        count = int(metric['count'])
        amount = int(metric['amount'])
        dis.append({'country': 'India','state': state_name,'year': year, 'quarter': 
quarter,'district_name': name,'transaction_count': count, 'total_amount': amount})

    df = pd.DataFrame(dis)
    dfs.append(df)
```

```
df_map_trans = pd.concat(dfs)
df_map_trans = df_map_trans.reset_index(drop=True)

df_map_trans.to_csv("C:\\Users\\Priyanka\\P!hu\\map_trans.csv")
print(df_map_trans)


#MAP_USER

import json
import pandas as pd
import glob
# specify the folder path using '*' pattern
path = 'C:\\Users\\Priyanka\\P!hu\\pulse\\data\\map\\user\\hover\\count
ry\\india\\state\\*\\*\\*.json'
files = glob.glob(path)

dfs = []

for file in files:
    folder = file.split("\\")
    country = folder[9]
    state_name = folder[12]
    year = int(folder[13])
    quarter_number = int(folder[14][0])
    quarter = "Q" + str(quarter_number)

    with open(file) as f:
        data = json.load(f)

    dis = []
    for states, users in data['data']['hoverData'].items():
        sate=states
        registered = int(users['registeredUsers'])
        app = int(users['appOpens'])
        dis.append({'country': 'India','state': state_name,'year': year, 'quarter':
quarter,'registered_users':registered,'apps_opened': app})

    df = pd.DataFrame(dis)
    dfs.append(df)

df_map_user = pd.concat(dfs)
df_map_user = df_map_user.reset_index(drop=True)

df_map_user.to_csv("C:\\Users\\Priyanka\\P!hu\\map_user.csv")
print(df_map_user)


#TOP_TRANSACTION

import json
import pandas as pd
import glob
# specify the folder path using '*' pattern
path ='C:\\Users\\Priyanka\\P!hu\\pulse\\data\\top\\transaction\\count
ry\\india\\state\\*\\*\\*.json'
files = glob.glob(path)

dfs = []

for file in files:
    folder = file.split("\\")
    state_name = folder[11]
    year = int(folder[12])
    quarter_number = int(folder[13][0])
    quarter = "Q" + str(quarter_number)

    with open(file) as f:
        data = json.load(f)

    dis = []
    for d in data['data']['districts']:
        name = d['entityName']
        metric = d['metric']
        count = int(metric['count'])
        amount = int(metric['amount'])
        dis.append({'country': 'India','state': state_name,'year': year, 'entity_type':
'district', 'quarter': quarter , 'district&pincode': name,'transaction_count': count,
'total_amount': amount})
    for pin in data['data']['pincodes']:
        name= pin['entityName']
        metric = pin['metric']
        count = int(metric['count'])
        amount = int(metric['amount'])
        dis.append({'country': 'India','state': state_name,'year': year, 'entity_type':
'pincode', 'quarter': quarter , 'district&pincode': name,'transaction_count': count,
'total_amount': amount})

    df = pd.DataFrame(dis)
    dfs.append(df)

df_top_trans = pd.concat(dfs)
df_top_trans = df_top_trans.reset_index(drop=True)

df_top_trans.to_csv("C:\\Users\\Priyanka\\P!hu\\top_trans.csv")
print(df_top_trans)

#TOP_USER

import json
import pandas as pd
import glob
# specify the folder path using '*' pattern
path ='C:\\Users\\Priyanka\\P!hu\\pulse\\data\\top\\user\\country\\india\\state\\*\\*\\*.json'
files = glob.glob(path)

dfs = []

for file in files:
    folder = file.split("\\")
    country = folder[8]
    state_name = folder[11]
    year = int(folder[12])
    quarter_number = int(folder[13][0])
    quarter = "Q" + str(quarter_number)

    with open(file) as f:
        data = json.load(f)

    dis = []
    for d in data['data']['districts']:
        dist = d['name']
        registered = int(d['registeredUsers'])
        dis.append({'country': 'India','state': state_name,'year': year, 'quarter': quarter,
'entity_type':'district','district&pin':dist,'registered_users': registered})
    for pin in data['data']['pincodes']:
        pincode = pin['name']
        registered = int(pin['registeredUsers'])
        dis.append({'country': 'India','state': state_name,'year': year, 'quarter': quarter,
'entity_type':'pincode','district&pin':pincode,'registered_users': registered})

    df = pd.DataFrame(dis)
    dfs.append(df)

df_top_user = pd.concat(dfs)
df_top_user = df_top_user.reset_index(drop=True)

df_top_user.to_csv("C:\\Users\\Priyanka\\P!hu\\top_user.csv")
print(df_top_user)


# using create_engine module opening the MySql with correct credentials
engine = create_engine('mysql+mysqlconnector://root:Pihu96@127.0.0.1:3306/pulse')
config = {
'user': 'root',
'password': 'Pihu96',
'host': '127.0.0.1',
'database': 'pulse',
'raise_on_warnings': True}

# Connect to the database
cnx = mysql.connector.connect(**config)

# Check if the connection is successful
if cnx.is_connected():
  print("Connection to MySQL database established.")
else:
  print("Connection to MySQL database failed.")

# create a table name and store the dataframe-1
df_agg_trans.to_sql(name='aggregate_transaction', con=engine, if_exists='replace',
index=False)
# create a table name and store the dataframe-2
df_agg_user.to_sql(name='aggregate_user', con=engine, if_exists='replace', index=False)
# create a table name and store the dataframe-3
df_map_trans.to_sql(name='map_transaction', con=engine, if_exists='replace',index=False)
# create a table name and store the dataframe-4
df_map_user.to_sql(name='map_user', con=engine, if_exists='replace', index=False)
# create a table name and store the dataframe-5
df_top_trans.to_sql(name='top_transaction', con=engine, if_exists='replace', index=False)
#create a table name and store the dataframe-6
df_top_user.to_sql(name='top_user', con=engine, if_exists='replace', index=False)

# # Create a cursor to execute SQL queries
cursor = cnx.cursor()

# Define the SQL query to retrieve data from the "table
query = "SELECT * FROM aggregate_transaction"

# Execute the SQL query and store the result in a Pandas dataframe
aggregate_transaction = pd.read_sql(query, cnx)

# Print the first 5 rows of the dataframe
```

```python
print(aggregate_transaction.head())

# SQL query to retrieve data from the "table
query = "SELECT * FROM aggregate_user"

# executing the SQL query and store the result in a Pandas dataframe
aggregate_user = pd.read_sql(query, cnx)

# Print the first 5 rows of the dataframe
print(aggregate_user.head())

# SQL query to retrieve data from the "table
query = "SELECT * FROM map_transaction"

# executing the SQL query and store the result in a Pandas dataframe
map_transaction = pd.read_sql(query, cnx)

# Print the first 5 rows of the dataframe
print(map_transaction.head())

# SQL query to retrieve data from the "table
query = "SELECT * FROM map_user"

# executing the SQL query and store the result in a Pandas dataframe
map_user = pd.read_sql(query, cnx)

# Print the first 5 rows of the dataframe
print(map_user.head())

# SQL query to retrieve data from the "table
query = "SELECT * FROM top_transaction"

# executing the SQL query and store the result in a Pandas dataframe
top_transaction = pd.read_sql(query, cnx)

# Print the first 5 rows of the dataframe
print(top_transaction.head())

# SQL query to retrieve data from the "table
query = "SELECT * FROM top_user"

# executing the SQL query and store the result in a Pandas dataframe
top_user = pd.read_sql(query, cnx)

# Print the first 5 rows of the dataframe
print(top_user.head())
```