In [1]:
```
import os
import json
import pandas as pd
```

In [2]:
```
# path to get aggregated transaction data
path1 = r'C:\Users\91994\Desktop\phonepe\pulse\data\aggregated\transaction\country\india\state'
# create an empty list to store data
data = []
# loop through directories
for st_dir in os.listdir(path1):
    st_path = os.path.join(path1, st_dir)
    for yr_dir in os.listdir(st_path):
        yr_path = os.path.join(st_path, yr_dir)
        for quart_file in os.listdir(yr_path):
            quart_path = os.path.join(yr_path, quart_file)
            with open(quart_path, 'r') as f:
                quart_data = json.load(f)
            for trans in quart_data['data']['transactionData']:
                name = trans['name']
                count = trans['paymentInstruments'][0]['count']
                amount = trans['paymentInstruments'][0]['amount']
                data.append({
                    'State': st_dir,
                    'Year': yr_dir,
                    'Quarter': int(os.path.splitext(quart_file)[0]),
                    'Transaction_type': name,
                    'Transaction_count': count,
                    'Transaction_amount': amount
                })
# create a dataframe from the data
df1_agg_trans = pd.DataFrame(data)
df1_agg_trans.head(5)
```

Out[2]:

	State	Year	Quarter	Transaction_type	Transaction_count	Transaction_amount
0	andaman-&-nicobar-islands	2018	1	Recharge & bill payments	4200	1.845307e+06
1	andaman-&-nicobar-islands	2018	1	Peer-to-peer payments	1871	1.213866e+07
2	andaman-&-nicobar-islands	2018	1	Merchant payments	298	4.525072e+05
3	andaman-&-nicobar-islands	2018	1	Financial Services	33	1.060142e+04
4	andaman-&-nicobar-islands	2018	1	Others	256	1.846899e+05

In [3]:
```
df1_agg_trans.to_csv('Agg_trans.csv',index = False)
```

In [4]:
```
# path to get aggregated user data
path2 = r'C:\Users\91994\Desktop\phonepe\pulse\data\aggregated\user\country\india\state'

# create an empty list to store data
data = []

# loop through directories
for st_dir in os.listdir(path2):
    st_path = os.path.join(path2, st_dir)
    for yr_dir in os.listdir(st_path):
        yr_path = os.path.join(st_path, yr_dir)
        for quart_file in os.listdir(yr_path):
            quart_path = os.path.join(yr_path, quart_file)
            with open(quart_path, 'r') as f:
                quart_data = json.load(f)
            try:
                for user in quart_data['data']['usersByDevice']:
                    brand = user['brand']
                    count = user['count']
                    percentage = user['percentage']
                    data.append({
                        'State': st_dir,
                        'Year': yr_dir,
                        'Quarter': int(os.path.splitext(quart_file)[0]),
                        'Brand': brand,
                        'Brand_count': count,
                        'Brand_percentage': percentage
                    })
            except:
                pass

# create a dataframe from the data
df2_agg_user = pd.DataFrame(data)
df2_agg_user.head(5)
```

Out[4]:

	State	Year	Quarter	Brand	Brand_count	Brand_percentage
0	andaman-&-nicobar-islands	2018	1	Xiaomi	1665	0.247033
1	andaman-&-nicobar-islands	2018	1	Samsung	1445	0.214392
2	andaman-&-nicobar-islands	2018	1	Vivo	982	0.145697
3	andaman-&-nicobar-islands	2018	1	Oppo	501	0.074332
4	andaman-&-nicobar-islands	2018	1	OnePlus	332	0.049258

In [5]:
```
df2_agg_user.to_csv('Agg_user.csv',index = False)
```

In [6]:
```
# path to get map transaction data
path3 = r'C:\Users\91994\Desktop\phonepe\pulse\data\map\transaction\hover\country\india\state'
# create an empty list to store data
data = []
# loop through directories
for st_dir in os.listdir(path3):
    st_path = os.path.join(path3, st_dir)
    for yr_dir in os.listdir(st_path):
        yr_path = os.path.join(st_path, yr_dir)
        for quart_file in os.listdir(yr_path):
            quart_path = os.path.join(yr_path, quart_file)
            with open(quart_path, 'r') as f :
                quart_data = json.load(f)
            for trans in quart_data['data']['hoverDataList']:
                district = trans['name']
                count = trans['metric'][0]['count']
                amount = trans['metric'][0]['amount']
                data.append({
                    'State': st_dir,
                    'Year': yr_dir,
                    'Quarter': int(os.path.splitext(quart_file)[0]),
                    'District': district,
                    'Transaction_count': count,
                    'Transaction_amount': amount
                })
# create a dataframe from the data
df3_map_trans = pd.DataFrame(data)
df3_map_trans.head(5)
```

Out[6]:

	State	Year	Quarter	District	Transaction_count	Transaction_amount
0	andaman-&-nicobar-islands	2018	1	north and middle andaman district	442	9.316631e+05
1	andaman-&-nicobar-islands	2018	1	south andaman district	5689	1.250025e+07
2	andaman-&-nicobar-islands	2018	1	nicobars district	528	1.139849e+06
3	andaman-&-nicobar-islands	2018	2	north and middle andaman district	825	1.317863e+06
4	andaman-&-nicobar-islands	2018	2	south andaman district	9395	2.394824e+07

In [7]:
```
df3_map_trans.to_csv('Map_trans.csv',index = False)
```

In [8]:
```
# path to get map user data
path4 = r'C:\Users\91994\Desktop\phonepe\pulse\data\map\user\hover\country\india\state'
# create an empty list to store data
data = []

# loop through directories
for st_dir in os.listdir(path4):
    st_path = os.path.join(path4, st_dir)
    for yr_dir in os.listdir(st_path):
        yr_path = os.path.join(st_path, yr_dir)
        for quart_file in os.listdir(yr_path):
            quart_path = os.path.join(yr_path, quart_file)
            with open(quart_path, 'r') as f:
                quart_data = json.load(f)
            for district in quart_data['data']['hoverData']:
                registered_user = quart_data['data']['hoverData'][district]["registeredUsers"]
                app_opening = quart_data['data']['hoverData'][district]["appOpens"]
                data.append({
                    'State': st_dir,
                    'Year': yr_dir,
                    'Quarter': int(os.path.splitext(quart_file)[0]),
                    'District': district,
                    'Registered_user': registered_user,
                    'App_opening': app_opening
                })

# create a dataframe from the data
df4_map_user = pd.DataFrame(data)
df4_map_user.head(5)
```

Out[8]:

	State	Year	Quarter	District	Registered_user	App_opening
0	andaman-&-nicobar-islands	2018	1	north and middle andaman district	632	0
1	andaman-&-nicobar-islands	2018	1	south andaman district	5846	0
2	andaman-&-nicobar-islands	2018	1	nicobars district	262	0
3	andaman-&-nicobar-islands	2018	2	north and middle andaman district	911	0
4	andaman-&-nicobar-islands	2018	2	south andaman district	8143	0

In [9]:
```
df4_map_user.to_csv('Map_user.csv',index = False)
```

In [10]:
```
# path to get top transaction data
path5 = r'C:\Users\91994\Desktop\phonepe\pulse\data\top\transaction\country\india\state'

# create an empty list to store data
data = []

# loop through directories
for st_dir in os.listdir(path5):
    st_path = os.path.join(path5, st_dir)
    for yr_dir in os.listdir(st_path):
        yr_path = os.path.join(st_path, yr_dir)
        for quart_file in os.listdir(yr_path):
            quart_path = os.path.join(yr_path, quart_file)
            with open(quart_path, 'r') as f:
                quart_data = json.load(f)
            for trans in quart_data['data']['pincodes']:
                name = trans['entityName']
                count = trans['metric']['count']
                amount = trans['metric']['amount']
                data.append({
                    'State': st_dir,
                    'Year': yr_dir,
                    'Quarter': int(os.path.splitext(quart_file)[0]),
                    'Pincode': name,
                    'Transaction_count': count,
                    'Transaction_amount': amount
                })

# create a dataframe from the data
df5_top_trans = pd.DataFrame(data)
df5_top_trans.head(5)
```

Out[10]:

	State	Year	Quarter	Pincode	Transaction_count	Transaction_amount
0	andaman-&-nicobar-islands	2018	1	744101	1622	2.769298e+06
1	andaman-&-nicobar-islands	2018	1	744103	1223	2.236042e+06
2	andaman-&-nicobar-islands	2018	1	744102	969	3.515906e+06
3	andaman-&-nicobar-islands	2018	1	744105	685	1.298561e+06
4	andaman-&-nicobar-islands	2018	1	744104	340	1.039715e+06

In [11]:
```
df5_top_trans.to_csv('Top_trans.csv',index = False)
```

In [12]:
```
# path to get top user data
path6 = r'C:\Users\91994\Desktop\phonepe\pulse\data\top\user\country\india\state'

# create an empty list to store data
data = []

# loop through state directories
for st_dir in os.listdir(path6):
    st_path = os.path.join(path6, st_dir)
    for yr_dir in os.listdir(st_path):
        yr_path = os.path.join(st_path, yr_dir)
        for quart_file in os.listdir(yr_path):
            quart_path = os.path.join(yr_path, quart_file)
            with open(quart_path, 'r') as f:
                quart_data = json.load(f)
            # loop through district data
            for district in quart_data['data']['pincodes']:
                name = district['name']
                users = district['registeredUsers']
                data.append({
                    'State': st_dir,
                    'Year': yr_dir,
                    'Quarter': int(os.path.splitext(quart_file)[0]),
                    'Pincode': name,
                    'Registered_user': users,
                })

# create a dataframe from the data
df6_top_user = pd.DataFrame(data)
df6_top_user.head(5)
```

Out[12]:

	State	Year	Quarter	Pincode	Registered_user
0	andaman-&-nicobar-islands	2018	1	744103	1608
1	andaman-&-nicobar-islands	2018	1	744101	1108
2	andaman-&-nicobar-islands	2018	1	744105	1075
3	andaman-&-nicobar-islands	2018	1	744102	1006
4	andaman-&-nicobar-islands	2018	1	744104	272

In [13]:
```
df6_top_user.to_csv('Top_user.csv',index = False)
```

In [14]:
```
import mysql.connector

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="Nilai23@"
)

mycursor = mydb.cursor(buffered=True)

mycursor.execute("CREATE DATABASE IF NOT EXISTS Phonepe")
mycursor.execute("USE Phonepe")
```

In [15]:
```
# Loop through each row in the DataFrame and insert it into the MySQL table

mycursor.execute('CREATE TABLE Agg_Transaction(Id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,State VARCHAR(100),Year INT,Quarter INT,Transaction_type VARCHAR(100),Transaction_count INT

for i, row in df1_agg_trans.iterrows():
    sql = "INSERT INTO Agg_Transaction (State, Year, Quarter, Transaction_type, Transaction_count, Transaction_amount) VALUES (%s, %s, %s, %s, %s, %s)"
    val = (row['State'], row['Year'], row['Quarter'], row['Transaction_type'], row['Transaction_count'], row['Transaction_amount'])

    mycursor.execute(sql, val)

# Commit the changes to the database
mydb.commit()
```

In [16]:
```
mycursor.execute('CREATE TABLE Agg_User(Id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,State VARCHAR(100),Year INT,Quarter INT, Brand VARCHAR(100),Brand_count INT,Brand_percentage FLOA

for i, row in df2_agg_user.iterrows():
    sql = "INSERT INTO Agg_User (State, Year, Quarter, Brand , Brand_count , Brand_percentage ) VALUES (%s, %s, %s, %s, %s, %s)"
    val = (row['State'], row['Year'], row['Quarter'], row['Brand'], row['Brand_count'], row['Brand_percentage'])

    mycursor.execute(sql, val)

# Commit the changes to the database
mydb.commit()
```

In [17]:
```
mycursor.execute('CREATE TABLE MAP_Transaction(Id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,State VARCHAR(100),Year INT,Quarter INT, district VARCHAR(100), Transaction_count INT, Tra

for i, row in df3_map_trans.iterrows():
    sql = "INSERT INTO MAP_Transaction (State, Year, Quarter, District , Transaction_count ,Transaction_amount ) VALUES (%s, %s, %s, %s, %s, %s)"
    val = (row['State'], row['Year'], row['Quarter'], row['District'], row['Transaction_count'], row['Transaction_amount'])

    mycursor.execute(sql, val)

# Commit the changes to the database
mydb.commit()
```

In [18]:
```
mycursor.execute('CREATE TABLE MAP_User(Id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,State VARCHAR(100),Year INT,Quarter INT, District VARCHAR(100),Registered_user INT, App_opening IN

for i, row in df4_map_user.iterrows():
    sql = "INSERT INTO MAP_User (State, Year, Quarter, District , Registered_user , App_opening ) VALUES (%s, %s, %s, %s, %s, %s)"
    val = (row['State'], row['Year'], row['Quarter'], row['District'], row['Registered_user'], row['App_opening'])

    mycursor.execute(sql, val)

# Commit the changes to the database
mydb.commit()
```

In [19]:
```
mycursor.execute('CREATE TABLE TOP_Transaction(Id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,State VARCHAR(100),Year INT,Quarter INT, Pincode INT, Transaction_count INT, Transaction_am

for i, row in df5_top_trans.iterrows():
    sql = "INSERT INTO TOP_Transaction (State, Year, Quarter, Pincode , Transaction_count ,Transaction_amount ) VALUES (%s, %s, %s, %s, %s, %s)"
    val = (row['State'], row['Year'], row['Quarter'], row['Pincode'], row['Transaction_count'], row['Transaction_amount'])

    mycursor.execute(sql, val)

# Commit the changes to the database
mydb.commit()
```

In [20]:
```
    mycursor.execute('CREATE TABLE TOP_User(Id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,State VARCHAR(100),Year INT,Quarter INT, Pincode INT,Registered_user INT)')

for i, row in df6_top_user.iterrows():
    sql = "INSERT INTO TOP_User (State, Year, Quarter, Pincode , Registered_user ) VALUES (%s, %s, %s, %s, %s)"
    val = (row['State'], row['Year'], row['Quarter'], row['Pincode'], row['Registered_user'])

    mycursor.execute(sql, val)

# Commit the changes to the database
mydb.commit()
```

In [ ]:
```

```

In [ ]:
```

```