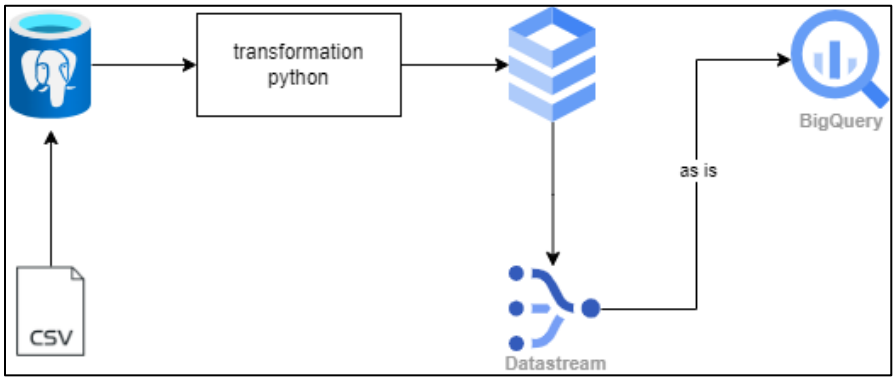
Link Google Cloud :

https://console.cloud.google.com/bigquery?referrer=search&project=glowing-patrol-418906&ws=!1m4!1m3!3m2!1sglowing-patrol-418906!2spublic

1. Flow



2. Data dari csv pada folder weekly assignment di load pada postgresQL local machine
   - Load data pada postgreSQL local machine
     - ✓ Import pandas dan psycopg2

```
import pandas as pd
import psycopg2
```
✓ 0.2s

     - ✓ Connect ke database

```
CONNECT_DB = "host=localhost port=5432 dbname=cloud_admin user=cloud_admin password=cloud_admin"
```
✓ 0.2s

       Sebelumnya membuat dulu database name, user, dan password cloud_admin di postgreSQL
     - ✓ Membuat table (tipe data belum sesuai karena data belum ditransformasi)
       - Untuk table bs140513_032310

```
create_table_query = '''CREATE TABLE bs140513_032310 (
    step int,
    customer varchar (20),
    age varchar (10),
    gender varchar (10),
    zipcodeOri varchar (20),
    merchant varchar (20),
    zipMerchant varchar (20),
    category varchar (30),
    amount float4,
    fraud boolean
); '''
```

- Untuk table bsnet140513_032310

```
create_table_query2 = '''CREATE TABLE bsnet140513_032310 (
    Source varchar (20),
    Target varchar (20),
    Weight float4,
    typeTrans varchar (30),
    fraud boolean
); '''
```

✓ Menambah data dari .csv table

- Untuk table bs140513_032310

```
try:
    # Make connection to db
    cxn = psycopg2.connect(CONNECT_DB)

    # Create a cursor to db
    cur = cxn.cursor()

    with open('./bs140513_032310.csv', 'r') as f:
        # skip first row, header row
        next(f)
        cur.copy_from(f, 'bs140513_032310', sep=",")
        cxn.commit()

except (Exception, psycopg2.Error) as error :
    print ("Error while connecting to PostgreSQL", error)

finally:
    #closing database connection.
    if(cxn):
        cur.close()
        cxn.close()
        print("PostgreSQL connection is closed")
        print("bs140513_032310 table populated")
```

- Untuk table bsnet140513_032310

```
try:
    # Make connection to db
    cxn2 = psycopg2.connect(CONNECT_DB)

    # Create a cursor to db
    cur2 = cxn2.cursor()

    with open('./bsnet140513_032310.csv', 'r') as f:
        # skip first row, header row
        next(f)
        cur2.copy_from(f, 'bsnet140513_032310', sep=",")
        cxn2.commit()

except (Exception, psycopg2.Error) as error :
    print ("Error while connecting to PostgreSQL", error)

finally:
    #closing database connection.
    if(cxn2):
        cur2.close()
        cxn2.close()
        print("PostgreSQL connection is closed")
        print("bsnet140513_032310 table populated")
```

- Setelah data berhasil di load pada postgresQL lalu masukkan data ke dalam dataframe seperti di bawah ini :
    - Untuk table bs140513_032310

Create dataframe

```
df = pd.DataFrame(records, columns=["step", "customer", "age", "gender",
"zipcodeOri", "merchant", "zipMerchant","category","amount","fraud"])
```

✓ 0.6s

    - Untuk table bsnet140513_032310

Create dataframe

```
df2 = pd.DataFrame(records2, columns=["Source", "Target", "Weight", "typeTrans", "fraud"])
```

✓ 0.5s

- Lalu lakukan EDA seperti mengecek tipe data, mengecek nilai null, melihat jumlah baris kolom untuk melihat dan memahami struktur data pada data frame.
    - Untuk table bs140513_032310

EDA (Exploratory Data Analysis)

```
df.info()
```

✓ 0.4s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 594643 entries, 0 to 594642
Data columns (total 10 columns):
 #   Column       Non-Null Count   Dtype  
---  ------       --------------   -----  
 0   step         594643 non-null  int64  
 1   customer     594643 non-null  object 
 2   age          594643 non-null  object 
 3   gender       594643 non-null  object 
 4   zipcodeOri   594643 non-null  object 
 5   merchant     594643 non-null  object 
 6   zipMerchant  594643 non-null  object 
 7   category     594643 non-null  object 
 8   amount       594643 non-null  float64
 9   fraud        594643 non-null  bool   
dtypes: bool(1), float64(1), int64(1), object(7)
memory usage: 41.4+ MB
```

```
df.dtypes
```

✓ 0.3s

```
step             int64
customer        object
age             object
gender          object
zipcodeOri      object
merchant        object
zipMerchant     object
category        object
amount         float64
fraud             bool
dtype: object
```

```python
#Check null
if df.isnull().values.any():
    print("Ada nilai null ")
else:
    print("Tidak ada nilai null")
```

✓ 0.7s

```
Tidak ada nilai null
```

```python
df.describe()
```

✓ 0.3s

	step	amount
count	594643.000000	594643.000000
mean	94.986827	37.890135
std	51.053632	111.402831
min	0.000000	0.000000
25%	52.000000	13.740000
50%	97.000000	26.900000
75%	139.000000	42.540000
max	179.000000	8329.960000

```python
df.head()
```

✓ 0.3s

	step	customer	age	gender	zipcodeOri	merchant	zipMerchant	category	amount	fraud
0	0	'C1093826151'	'4'	'M'	'28007'	'M348934600'	'28007'	'es_transportation'	4.55	False
1	0	'C352968107'	'2'	'M'	'28007'	'M348934600'	'28007'	'es_transportation'	39.68	False
2	0	'C2054744914'	'4'	'F'	'28007'	'M1823072687'	'28007'	'es_transportation'	26.89	False
3	0	'C1760612790'	'3'	'M'	'28007'	'M348934600'	'28007'	'es_transportation'	17.25	False
4	0	'C757503768'	'5'	'M'	'28007'	'M348934600'	'28007'	'es_transportation'	35.72	False

```
df.tail()
```
✓ 0.2s

	step	customer	age	gender	zipcodeOri	merchant	zipMerchant	category	amount	fraud
594638	179	'C1753498738'	'3'	'F'	'28007'	'M1823072687'	'28007'	'es_transportation'	20.53	False
594639	179	'C650108285'	'4'	'F'	'28007'	'M1823072687'	'28007'	'es_transportation'	50.73	False
594640	179	'C123623130'	'2'	'F'	'28007'	'M349281107'	'28007'	'es_fashion'	22.44	False
594641	179	'C1499363341'	'5'	'M'	'28007'	'M1823072687'	'28007'	'es_transportation'	14.46	False
594642	179	'C616528518'	'4'	'F'	'28007'	'M1823072687'	'28007'	'es_transportation'	26.93	False

```
df.count()
```
✓ 0.6s

```
step          594643
customer      594643
age           594643
gender        594643
zipcodeOri    594643
merchant      594643
zipMerchant   594643
category      594643
amount        594643
fraud         594643
dtype: int64
```

- Untuk table bs140513_032310

EDA (Exploratory Data Analysis)

```
df2.head()
```

✓ 0.1s

	Source	Target	Weight	typeTrans	fraud
0	'C1093826151'	'M348934600'	4.55	'es_transportation'	False
1	'C352968107'	'M348934600'	39.68	'es_transportation'	False
2	'C2054744914'	'M1823072687'	26.89	'es_transportation'	False
3	'C1760612790'	'M348934600'	17.25	'es_transportation'	False
4	'C757503768'	'M348934600'	35.72	'es_transportation'	False

```
df2.tail()
```

✓ 0.6s

	Source	Target	Weight	typeTrans	fraud
594638	'C1753498738'	'M1823072687'	20.53	'es_transportation'	False
594639	'C650108285'	'M1823072687'	50.73	'es_transportation'	False
594640	'C123623130'	'M349281107'	22.44	'es_fashion'	False
594641	'C1499363341'	'M1823072687'	14.46	'es_transportation'	False
594642	'C616528518'	'M1823072687'	26.93	'es_transportation'	False

```
df2.info()
```

✓ 0.2s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 594643 entries, 0 to 594642
Data columns (total 5 columns):
 #   Column     Non-Null Count   Dtype
---  ------     --------------   -----
 0   Source     594643 non-null  object
 1   Target     594643 non-null  object
 2   Weight     594643 non-null  float64
 3   typeTrans  594643 non-null  object
 4   fraud      594643 non-null  bool
dtypes: bool(1), float64(1), object(3)
memory usage: 18.7+ MB
```

```python
#Check null
if df2.isnull().values.any():
    print("Ada nilai null di df2")
else:
    print("Tidak ada nilai null di df2")
```

✓ 0.2s

```
Tidak ada nilai null di df2
```

```python
df2.describe()
```

✓ 0.1s

	Weight
count	594643.000000
mean	37.890135
std	111.402831
min	0.000000
25%	13.740000
50%	26.900000
75%	42.540000
max	8329.960000

```python
df2.count()
```

✓ 0.2s

```
Source       594643
Target       594643
Weight       594643
typeTrans    594643
fraud        594643
dtype: int64
```

3. Transformasi:
   A. Untuk data pada table bs140513_032310
   i. Menghilangkan tanda petik (' ') pada beberapa kolom, antara lain : customer, age, gender, zipcodeOri, merchant, zipMerchant, category.

   Alasan : karena pada table bs140513_032310 pada kolom customer, age, gender, zipcodeOri, merchant, zipMerchant, dan category terdapat tanda (' '). Dari kolom-kolom tersebut tidak memerlukan karakter khusus yang memerlukan tanda petik (' ') sehingga tanda petik dihapus.

   Dengan code sebagai berikut :

```
clean_columns = ['customer', 'age', 'gender', 'zipcodeOri', 'merchant', 'zipMerchant', 'category']
for col in clean_columns:
    df[col] = df[col].str.replace("'", "")

#Show 5 data
df.head()
```

✓ 1.5s

	step	customer	age	gender	zipcodeOri	merchant	zipMerchant	category	amount	fraud
0	0	C1093826151	4	M	28007	M348934600	28007	es_transportation	4.55	False
1	0	C352968107	2	M	28007	M348934600	28007	es_transportation	39.68	False
2	0	C2054744914	4	F	28007	M1823072687	28007	es_transportation	26.89	False
3	0	C1760612790	3	M	28007	M348934600	28007	es_transportation	17.25	False
4	0	C757503768	5	M	28007	M348934600	28007	es_transportation	35.72	False

   ii. Mengganti data U yang ada pada kolom age
   Alasan : karena setelah menghapus tanda baca (‘ ‘) pada kolom age & ingin mengubah tipe data age menjadi integer ternyata masih belum bisa dan ditemukan error seperti berikut :

```
ValueError: invalid literal for int() with base 10: 'U'
```

   Sehingga karakter U yang ada pada kolom age diganti dengan data rata-rata umur

Replace U with average age in the age column

```
#Count average age from valid data
mean_age = df[df['age'] != 'U']['age'].astype(int).mean()
```

✓ 0.3s

```
#Replace 'U' with average age
df.loc[df['age'] == 'U', 'age'] = mean_age
```

✓ 0.1s

Lalu mengganti tipe data kolom age menjadi INTEGER

```
#Change data type for age to integer
df['age'] = df['age'].astype(int)
```

✓ 0.1s

```
df.info()
```

✓ 0.3s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 594643 entries, 0 to 594642
Data columns (total 10 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   step         594643 non-null  int64
 1   customer     594643 non-null  object
 2   age          594643 non-null  int32
 3   gender       594643 non-null  object
 4   zipcodeOri   594643 non-null  object
 5   merchant     594643 non-null  object
 6   zipMerchant  594643 non-null  object
 7   category     594643 non-null  object
 8   amount       594643 non-null  float64
 9   fraud        594643 non-null  bool
dtypes: bool(1), float64(1), int32(1), int64(1), object(6)
memory usage: 39.1+ MB
```

iii. Mengubah tipe data pada beberapa kolom, antara lain : zipcodeOri dan zipMerchant menjadi INTEGER

Alasan : karena pada dataset/table bs140513_032310 zipcodeOri dan zipMerchant awalnya dimasukkan dengan tipe data VARCHAR karena terdapat tanda petik (‘’) tetapi karena tanda tersebut sudah dihapis sehingga pada kolom tersebut hanya berisikan data kombinasi angka (bilangan bulat) maka tipe data diubah menjadi INTEGER.

Dengan code sebagai berikut :

Change data type zipcodeOri and zipMerchant to integer

```
df['zipcodeOri'] = df['zipcodeOri'].astype(int)
df['zipMerchant'] = df['zipMerchant'].astype(int)
```

✓ 0.1s

```
df.dtypes
```

✓ 0.1s

```
step             int64
customer        object
age              int32
gender          object
zipcodeOri       int32
merchant        object
zipMerchant      int32
category        object
amount         float64
fraud             bool
dtype: object
```

iv. Casefolding : mengubah data yang terdapat pada kolom customer, gender, dan merchant menjadi huruf kecil semua (lowercase)

Alasan : agar karakter pada data sama semua (seragam) dan memudahkan proses analisis data.

Data awal :

	step	customer	age	gender	zipcodeOri	merchant	zipMerchant	category	amount	fraud
0	0	C1093826151	4	M	28007	M348934600	28007	es_transportation	4.55	False
1	0	C352968107	2	M	28007	M348934600	28007	es_transportation	39.68	False
2	0	C2054744914	4	F	28007	M1823072687	28007	es_transportation	26.89	False
3	0	C1760612790	3	M	28007	M348934600	28007	es_transportation	17.25	False
4	0	C757503768	5	M	28007	M348934600	28007	es_transportation	35.72	False

Code untuk mengubah menjadi lowercase :

Casefolding : Transform data in the columns customer, gender, and merchant into lowercase

```
df['customer'] = df['customer'].str.lower()
df['gender'] = df['gender'].str.lower()
df['merchant'] = df['merchant'].str.lower()
```

✓ 1.2s

v. Mengubah nama kolom yang masih ada huruf kapital (zipcodeOri dan zipMerchant) menjadi huruf kecil semua (lowercase)

Alasan : agar karakter pada data sama semua (seragam) dan memudahkan proses analisis data.
Dengan code seperti ini :

```
Change columns name, zipcodeOri to zipcodeori and zipMerchant to zipmerchant (lowercase)

df.rename(columns={"zipcodeOri": "zipcodeori", "zipMerchant": "zipmerchant"}, inplace=True)
✓ 0.6s
```

Hasil :

```
df.head()
✓ 0.1s
```

	step	customer	age	gender	zipcodeori	merchant	zipmerchant	category	amount	fraud
0	0	c1093826151	4	m	28007	m348934600	28007	es_transportation	4.55	False
1	0	c352968107	2	m	28007	m348934600	28007	es_transportation	39.68	False
2	0	c2054744914	4	f	28007	m1823072687	28007	es_transportation	26.89	False
3	0	c1760612790	3	m	28007	m348934600	28007	es_transportation	17.25	False
4	0	c757503768	5	m	28007	m348934600	28007	es_transportation	35.72	False

vi. Pengecekan nilai NULL dengan df.count() (menghitung jumlah nilai non null), karena tidak ada nilai null sehingga tidak dilakukan proses filtering missing values.

```
df.count()
✓ 0.6s

step          594643
customer      594643
age           594643
gender        594643
zipcodeori    594643
merchant      594643
zipmerchant   594643
category      594643
amount        594643
fraud         594643
dtype: int64
```

```
if df.isnull().values.any():
    print("Ada nilai null")
else:
    print("Tidak ada nilai null")
```
✓ 0.2s

```
Tidak ada nilai null
```

Lalu simpan data ke dalam file .csv

Save dataframe to csv file

```
df.to_csv("bs_clean.csv", index=False, header=False)
```
✓ 3.2s

Hasil akhir (5 baris pertama data pada table bs140513_032310):

Sebelum transformasi data :

```
df.head()
```
✓ 0.3s

	step	customer	age	gender	zipcodeOri	merchant	zipMerchant	category	amount	fraud
0	0	'C1093826151'	'4'	'M'	'28007'	'M348934600'	'28007'	'es_transportation'	4.55	False
1	0	'C352968107'	'2'	'M'	'28007'	'M348934600'	'28007'	'es_transportation'	39.68	False
2	0	'C2054744914'	'4'	'F'	'28007'	'M1823072687'	'28007'	'es_transportation'	26.89	False
3	0	'C1760612790'	'3'	'M'	'28007'	'M348934600'	'28007'	'es_transportation'	17.25	False
4	0	'C757503768'	'5'	'M'	'28007'	'M348934600'	'28007'	'es_transportation'	35.72	False

Sesudah transformasi data :

```
df.head()
```
✓ 0.1s

	step	customer	age	gender	zipcodeori	merchant	zipmerchant	category	amount	fraud
0	0	c1093826151	4	m	28007	m348934600	28007	es_transportation	4.55	False
1	0	c352968107	2	m	28007	m348934600	28007	es_transportation	39.68	False
2	0	c2054744914	4	f	28007	m1823072687	28007	es_transportation	26.89	False
3	0	c1760612790	3	m	28007	m348934600	28007	es_transportation	17.25	False
4	0	c757503768	5	m	28007	m348934600	28007	es_transportation	35.72	False

B. Data pada bsnet140513_032310.csv

5 baris pertama data pada dataframe/table bsnet140513_032310

```
df2.head()
```
✓ 0.1s

	Source	Target	Weight	typeTrans	fraud
0	'C1093826151'	'M348934600'	4.55	'es_transportation'	0
1	'C352968107'	'M348934600'	39.68	'es_transportation'	0
2	'C2054744914'	'M1823072687'	26.89	'es_transportation'	0
3	'C1760612790'	'M348934600'	17.25	'es_transportation'	0
4	'C757503768'	'M348934600'	35.72	'es_transportation'	0

i. Menghilangkan tanda petik (' ') pada beberapa kolom, antara lain : Source, Target, typeTrans.

Alasan : karena pada dataframe/table bs140513_032310 pada kolom Source, Target, dan typeTrans terdapat tanda (' '). Dari kolom-kolom tersebut tidak memerlukan karakter khusus yang memerlukan tanda petik (' ') sehingga tanda petik dihapus.

Dengan code sebagai berikut :

```
clean_columns_df2 = ['Source', 'Target', 'typeTrans']
for col in clean_columns_df2:
    df2[col] = df2[col].str.replace("'", "")

#Show 5 data
df2.head()
```
✓ 0.7s

Hasil :

	Source	Target	Weight	typeTrans	fraud
0	C1093826151	M348934600	4.55	es_transportation	0
1	C352968107	M348934600	39.68	es_transportation	0
2	C2054744914	M1823072687	26.89	es_transportation	0
3	C1760612790	M348934600	17.25	es_transportation	0
4	C757503768	M348934600	35.72	es_transportation	0

ii. Casefolding : mengubah nama kolom Source, Targer, dan Weight menjadi huruf kecil semua (lowercase)

Alasan : agar karakter pada data sama semua (seragam) dan memudahkan proses analisis data.

Code untuk mengubah menjadi lowercase :

```
Change columns name into lowercase : Source -> source, Target->targer, Weight -> weight, typeTrans->typetrans

df2.rename(columns={"Source": "source", "Target": "target", 'Weight': 'weight', 'typeTrans' : 'typetrans'}, inplace=True)
✓ 0.2s
```

Hasil :

	source	target	weight	typetrans	fraud
0	C1093826151	M348934600	4.55	es_transportation	False
1	C352968107	M348934600	39.68	es_transportation	False
2	C2054744914	M1823072687	26.89	es_transportation	False
3	C1760612790	M348934600	17.25	es_transportation	False
4	C757503768	M348934600	35.72	es_transportation	False

iii. Casefolding : mengubah data yang terdapat pada kolom source dan target menjadi huruf kecil semua (lowercase)

Alasan : agar karakter pada data sama semua (seragam) dan memudahkan proses analisis data.

Dengan code seperti ini :

```
Casefolding : Changes data in the source and target columns into lowercase

df2['source'] = df2['source'].str.lower()
df2['target'] = df2['target'].str.lower()
✓ 0.9s
```

Hasil :

	source	target	weight	typetrans	fraud
0	c1093826151	m348934600	4.55	es_transportation	False
1	c352968107	m348934600	39.68	es_transportation	False
2	c2054744914	m1823072687	26.89	es_transportation	False
3	c1760612790	m348934600	17.25	es_transportation	False
4	c757503768	m348934600	35.72	es_transportation	False

iv. Pengecekan nilai NULL dengan df.count() (menghitung jumlah nilai non null), karena tidak ada nilai null sehingga tidak dilakukan proses filtering missing values.

```
df2.count()
```
✓ 0.2s

```
source       594643
target       594643
weight       594643
typetrans    594643
fraud        594643
dtype: int64
```

```
if df2.isnull().values.any():
    print ("ada nilai null")
else:
    print ("tidak ada nilai null")
```
✓ 0.5s

```
tidak ada nilai null
```

Setelah itu simpan data ke dalam .csv

Save dataframe to csv file

```
df2.to_csv('bsnet_clean.csv', index=False, header=False)
```
✓ 2.1s

Hasil akhir :

5 baris pertama data pada dataframe/table bsnet140513_032310
Sebelum transformasi :

	Source	Target	Weight	typeTrans	fraud
0	'C1093826151'	'M348934600'	4.55	'es_transportation'	False
1	'C352968107'	'M348934600'	39.68	'es_transportation'	False
2	'C2054744914'	'M1823072687'	26.89	'es_transportation'	False
3	'C1760612790'	'M348934600'	17.25	'es_transportation'	False
4	'C757503768'	'M348934600'	35.72	'es_transportation'	False

Sesudah transformasi :

	source	target	weight	typetrans	fraud
0	c1093826151	m348934600	4.55	es_transportation	False
1	c352968107	m348934600	39.68	es_transportation	False
2	c2054744914	m1823072687	26.89	es_transportation	False
3	c1760612790	m348934600	17.25	es_transportation	False
4	c757503768	m348934600	35.72	es_transportation	False

4. Setelah datanya bersih, atau sesuai yang dikehendaki, lakukan load ke Cloud SQL di google cloud platform dengan nama table yang sama
   - Inisialisasi (koneksi local ke google cloud) dengan code gcloud init, setelah itu masuk ke akun yang akan dipakai

```
C:\Program Files (x86)\Google\Cloud SDK>gcloud init
Welcome! This command will take you through the configuration of gcloud.

Settings from your current configuration [default] are:
accessibility:
  screen_reader: 'False'
core:
  account: ivanapuspitasari11@gmail.com
  disable_usage_reporting: 'True'
  project: glowing-patrol-418906

Pick configuration to use:
 [1] Re-initialize this configuration [default] with new settings
 [2] Create a new configuration
 [3] Switch to and re-initialize existing configuration: [region]
Please enter your numeric choice:  1

Your current configuration has been set to: [default]
```

- Setelah masuk ke akun yang akan dipakai lalu pilih project yang akan digunakan

```
You are logged in as: [ivanapuspitasari11@gmail.com].

Pick cloud project to use:
 [1] data-fellowship12
 [2] glowing-patrol-418906
 [3] Enter a project ID
 [4] Create a new project
Please enter numeric choice or text value (must exactly match list item):  2

Your current project has been set to: [glowing-patrol-418906].
```

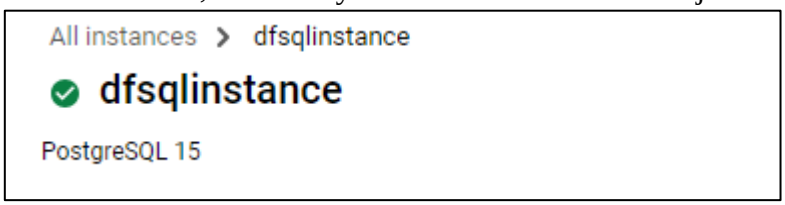
- Membuat bucket dengan cara sebagai berikut :

```
C:\Program Files (x86)\Google\Cloud SDK>gsutil mb gs://assignment_bucket_df
Creating gs://assignment_bucket_df/...

C:\Program Files (x86)\Google\Cloud SDK>gsutil ls
gs://assignment_bucket_df/
gs://df12_test_bucket_python_client/
gs://testing_bucket_df/
```

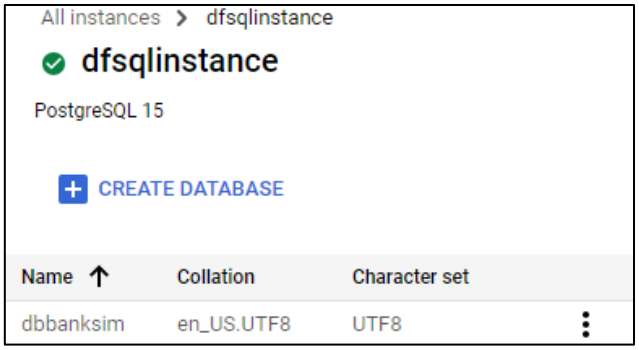
- Masukkan data ke bucket dengan cara sebagai berikut (untuk data bs_clean.csv dan bsnet_clean.csv :

```
C:\Program Files (x86)\Google\Cloud SDK>gcloud storage cp D:/datafellowship12/week1
assignment/bs_clean.csv gs://assignment_bucket_df/
Copying file://D:\datafellowship12\week1assignment\bs_clean.csv to gs://assignment_
bucket_df/bs_clean.csv
  Completed files 1/1 | 41.0MiB/41.0MiB | 1.8MiB/s

Average throughput: 3.1MiB/s

C:\Program Files (x86)\Google\Cloud SDK>gcloud storage cp D:/datafellowship12/week1
assignment/bsnet_clean.csv gs://assignment_bucket_df/
Copying file://D:\datafellowship12\week1assignment\bsnet_clean.csv to gs://assignme
nt_bucket_df/bsnet_clean.csv
  Completed files 1/1 | 30.0MiB/30.0MiB | 3.8MiB/s

Average throughput: 3.8MiB/s

C:\Program Files (x86)\Google\Cloud SDK>
```

- Create Instance, karena saya sudah membuat instance jadi memakai yang sudah ada



- Create database (dbbanksim)



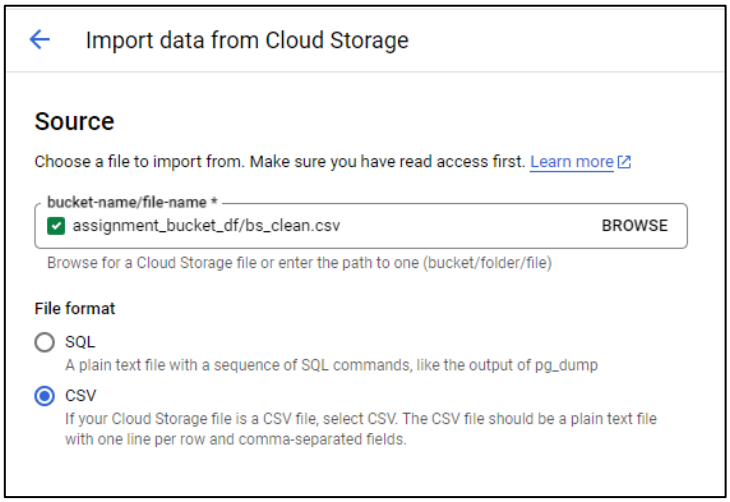
- Buat table
  - ✓ Table untuk bs_clean dengan query seperti di bawah ini :

```
dbbanksim=> CREATE TABLE bs_clean (
    step int,
    customer varchar (20),
    age int,
    gender varchar (10),
    zipcodeori int,
    merchant varchar (20),
    zipmerchant int,
    category varchar (30),
    amount float4,
    fraud boolean
);
CREATE TABLE
dbbanksim=> \dt
          List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | bs_clean | table | postgres
(1 row)

dbbanksim=>
```

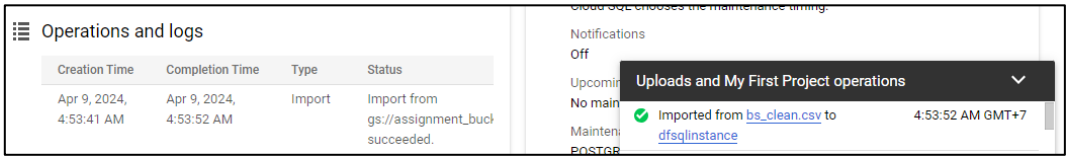
  - ✓ Table untuk bsnet_clean dengan query seperti di bawah ini :

```
dbbanksim=> CREATE TABLE bsnet_clean (
    source varchar (20),
    target varchar (20),
    weight float4,
    typetrans varchar (30),
    fraud boolean
);
CREATE TABLE
dbbanksim=> \dt
            List of relations
 Schema |    Name     | Type  |  Owner
--------+-------------+-------+----------
 public | bs_clean    | table | postgres
 public | bsnet_clean | table | postgres
(2 rows)

dbbanksim=>
```

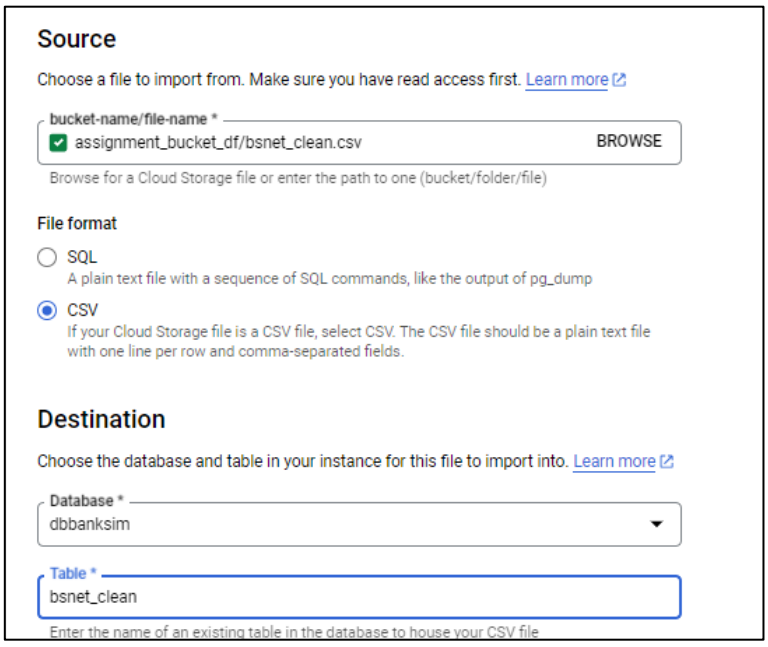
- Import data dari cloud storage
  - ✓ Untuk data dari bs_clean.csv



  - ✓ Data berhasil di import



  - ✓ Untuk data dari bsnet_clean.csv



  - ✓ Data berhasil di import

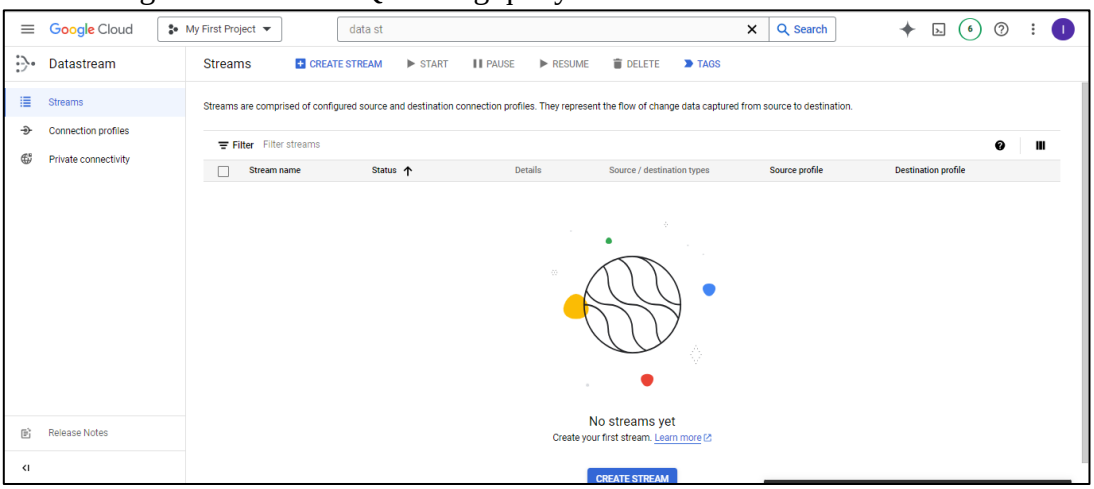
- Cek data apakah sudah masuk ke Cloud SQL atau belum dengan query di bawah ini :
  - ✓ Untuk table bs_clean

```
dbbanksim=>
dbbanksim=> SELECT * FROM bs_clean LIMIT 5;
 step |  customer   | age | gender | zipcodeori |  merchant   | zipmerchant |     category      | amount | fraud
------+-------------+-----+--------+------------+-------------+-------------+-------------------+--------+-------
    0 | c1093826151 |   4 | m      |      28007 | m348934600  |       28007 | es_transportation |   4.55 | f
    0 | c352968107  |   2 | m      |      28007 | m348934600  |       28007 | es_transportation |  39.68 | f
    0 | c2054744914 |   4 | f      |      28007 | m1823072687 |       28007 | es_transportation |  26.89 | f
    0 | c1760612790 |   3 | m      |      28007 | m348934600  |       28007 | es_transportation |  17.25 | f
    0 | c757503768  |   5 | m      |      28007 | m348934600  |       28007 | es_transportation |  35.72 | f
(5 rows)

dbbanksim=>
```

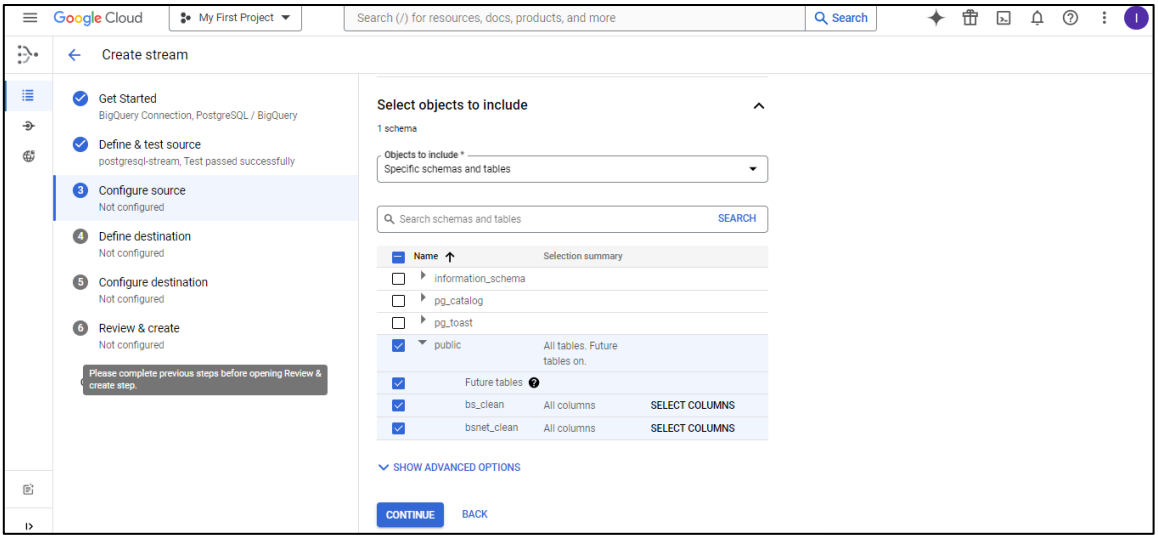
  - ✓ Untuk table bsnet_clean

```
dbbanksim=> SELECT * FROM bsnet_clean LIMIT 5;
   source    |   target    | weight |     typetrans     | fraud
-------------+-------------+--------+-------------------+-------
 c1093826151 | m348934600  |   4.55 | es_transportation | f
 c352968107  | m348934600  |  39.68 | es_transportation | f
 c2054744914 | m1823072687 |  26.89 | es_transportation | f
 c1760612790 | m348934600  |  17.25 | es_transportation | f
 c757503768  | m348934600  |  35.72 | es_transportation | f
(5 rows)

dbbanksim=>
```

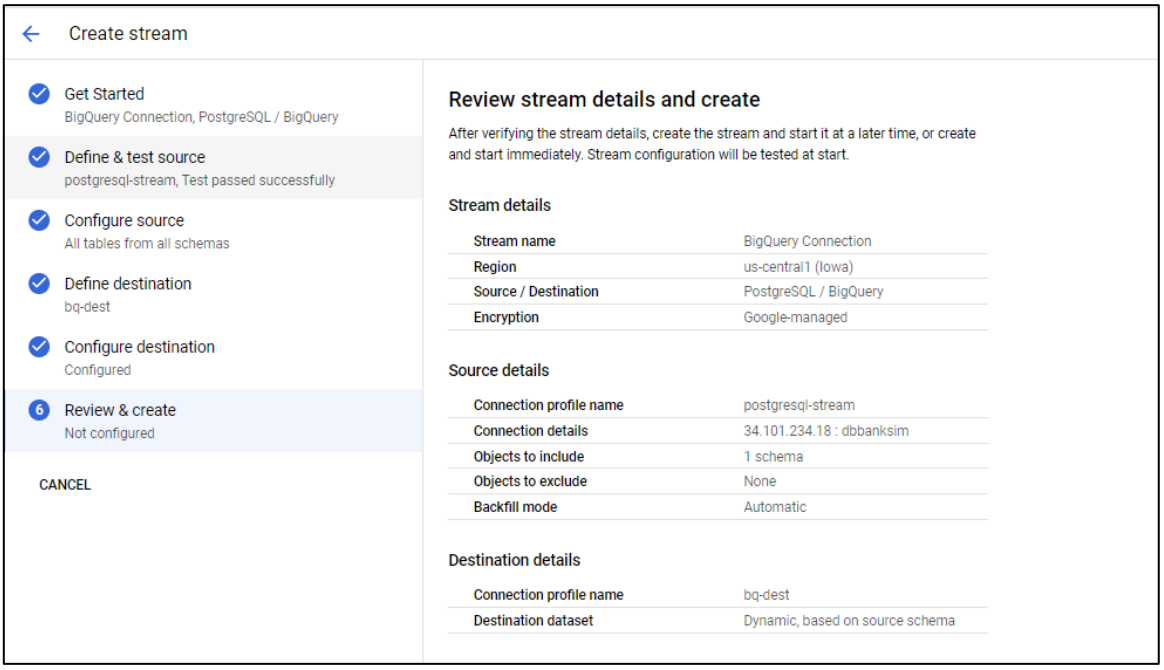
5. Setelah datanyanya landing pada Cloud SQL, offload ke BigQuery secara as is atau apa adannya
   - Landing data ke cloud SQL ke big query melalui datasream



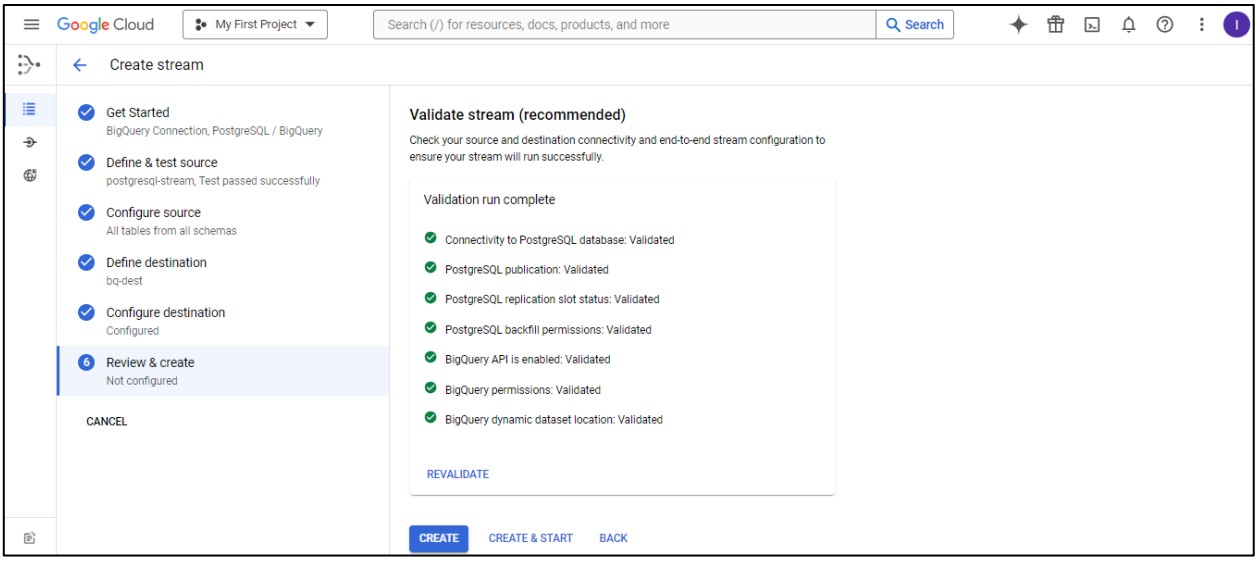
- Ikuti 6 langkah berikut



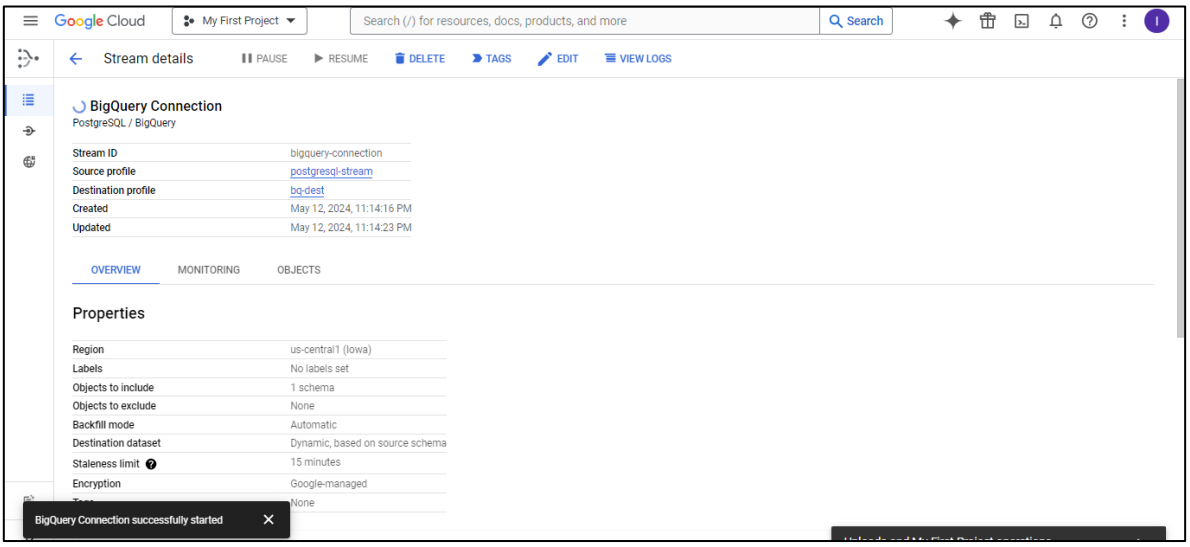
- Apabila sudah sampai langkah ke 6, validate konfigurasinya



- Seteah sudah tervalidasi semua, pencet “create and start”



- Tunggu sampai data landing di BigQuery



- Hasil dataset di BigQuery