import numpy as np

import pandas as pd

data = pd.read\_csv("Master\_Table - MasterTable\_export.csv", low\_memory=False)

data.head(10)  # Display the first 10 rows of the DataFrame to check the data structure

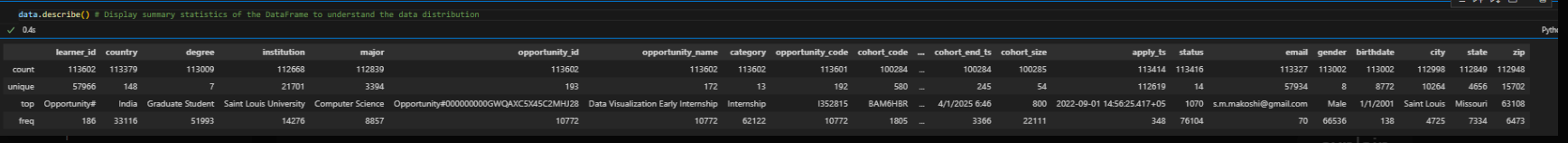
1.initially (rows and columns)

data.shape # Display the shape of the DataFrame to understand its dimensions

(113602, 21)

2. Initial statistics of master table with data type – Object

data.describe() # Display summary statistics of the DataFrame to understand the data distribution



**1. Count**

Number of non-null (non-missing) entries per column.  
Example:

* learner\_id has 113,602 values (so a few missing, since your dataset is bigger).
* degree has 113,009 values → some missing.

**2. Unique**

Number of distinct values in that column.  
Example:

* country has 148 unique values.
* degree has 7 unique values.
* opportunity\_id has 193 unique IDs.

**3. Top**

The most frequently occurring value in that column.  
Example:

* For country, the most common is **India**.
* For degree, the most common is **Graduate Student**.
* For opportunity\_name, the most common is **Data Visualization Early Internship**.

**4. Freq**

How many times the “top” value occurs.  
Example:

* Graduate Student occurs 51,993 times.
* India appears 33,116 times.
* Saint Louis University appears 14,276 times.

**Key Insights:**

* Dataset has **113k+ rows**.
* Some missing values (since counts differ per column).
* Categorical columns like degree, country, institution, gender have dominant categories.
* zip, state, city also categorical.
* cohort\_size, birthdate look like numeric/date columns but are being treated as strings — might need conversion
* # Display data types of all columns
* print(data.dtypes)
* # (Optional) Show number of unique types per column
* print("\nColumn-wise data type counts:")
* print(data.dtypes.value\_counts())

learner\_id object

country object

degree object

institution object

major object

opportunity\_id object

opportunity\_name object

category object

opportunity\_code object

cohort\_code object

cohort\_start\_ts object

cohort\_end\_ts object

cohort\_size object

apply\_ts object

status object

email object

gender object

birthdate object

city object

state object

zip object

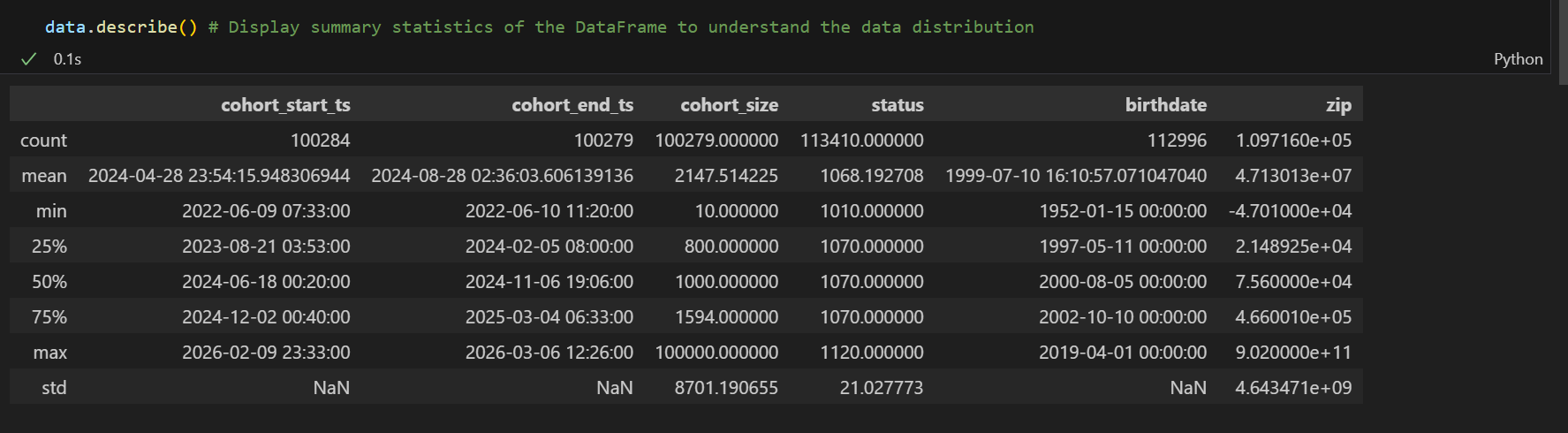
dtype: object

Column-wise data type counts:

object 21

Name: count, dtype: int64

statistics of master table after changing data type of columns– string ,date and float



# Display data types of all columns

print(data.dtypes)

# (Optional) Show number of unique types per column

print("\nColumn-wise data type counts:")

print(data.dtypes.value\_counts())

learner\_id string[python]

country string[python]

degree string[python]

institution string[python]

major string[python]

opportunity\_id string[python]

opportunity\_name string[python]

category string[python]

opportunity\_code string[python]

cohort\_code string[python]

cohort\_start\_ts datetime64[ns]

cohort\_end\_ts datetime64[ns]

cohort\_size float64

apply\_ts datetime64[ns, UTC+05:00]

status float64

email string[python]

gender string[python]

birthdate datetime64[ns]

city string[python]

state string[python]

zip float64

dtype: object

Column-wise data type counts:

string[python] 14

datetime64[ns] 3

float64 3

datetime64[ns, UTC+05:00] 1

Name: count, dtype: int64

# 1. Convert selected numeric columns

numeric\_cols = ["cohort\_size", "status", "zip"]

for col in numeric\_cols:

    data[col] = pd.to\_numeric(data[col], errors="coerce")

# 2. Convert selected datetime columns

date\_cols = ["apply\_ts", "birthdate", "cohort\_start\_ts", "cohort\_end\_ts"]

for col in date\_cols:

    data[col] = pd.to\_datetime(data[col], errors="coerce")

# Find all categorical columns

cat\_cols = data.select\_dtypes(include=["category"]).columns

# Convert them to pandas string dtype

data[cat\_cols] = data[cat\_cols].astype("string")

# Check updated datatypes

print(data.dtypes)

# Summary for numeric + datetime

print("\nNumeric Summary:")

print(data[numeric\_cols].describe())

# Summary for datetime columns (custom summary)

print("\nDatetime Summary:")

for col in date\_cols:

    print(f"\nColumn: {col}")

    print("Min:", data[col].min())

    print("Max:", data[col].max())

    print("Non-Null Count:", data[col].notnull().sum())

Numeric Summary:

cohort\_size status zip

count 100279.000000 113410.000000 1.097160e+05

mean 2147.514225 1068.192708 4.713013e+07

std 8701.190655 21.027773 4.643471e+09

min 10.000000 1010.000000 -4.701000e+04

25% 800.000000 1070.000000 2.148925e+04

50% 1000.000000 1070.000000 7.560000e+04

75% 1594.000000 1070.000000 4.660010e+05

max 100000.000000 1120.000000 9.020000e+11

Datetime Summary:

Column: apply\_ts

Min: 2022-06-09 16:28:33.977000+05:00

Max: 2025-02-25 05:15:42.257000+05:00

Non-Null Count: 113292

Column: birthdate

Min: 1952-01-15 00:00:00

Max: 2019-04-01 00:00:00

Non-Null Count: 112996

Column: cohort\_start\_ts

Min: 2022-06-09 07:33:00

Max: 2026-02-09 23:33:00

Non-Null Count: 100284

Column: cohort\_end\_ts

Min: 2022-06-10 11:20:00

Max: 2026-03-06 12:26:00

Non-Null Count: 100279

learner\_id country degree institution \

count 113602 113379 113009 112668

unique 57966 148 7 21701

top Opportunity# India Graduate Student Saint Louis University

freq 186 33116 51993 14276

major opportunity\_id \

count 112839 113602

unique 3394 193

top Computer Science Opportunity#000000000GWQAXC5X45C2MHJ28

freq 8857 10772

opportunity\_name category opportunity\_code \

count 113602 113602 113601

unique 172 13 192

top Data Visualization Early Internship Internship I352815

freq 10772 62122 10772

cohort\_code email gender city state

count 100284 113327 113002 112998 112849

unique 580 57934 8 10264 4656

top BAM6HBR s.m.makoshi@gmail.com Male Saint Louis Missouri

freq 1805 70 66536 4725 7334

ETL STARTS HERE

#all columns

columns =list(data.columns)

print("\nAll Columns:")

columns

['learner\_id',

'country',

'degree',

'institution',

'major',

'opportunity\_id',

'opportunity\_name',

'category',

'opportunity\_code',

'cohort\_code',

'cohort\_start\_ts',

'cohort\_end\_ts',

'cohort\_size',

'apply\_ts',

'status',

'email',

'gender',

'birthdate',

'city',

'state',

'zip']

**Checking null values before counting zero**

data.isnull().sum()  # Check for missing values in each column

learner\_id 0

country 223

degree 593

institution 934

major 763

opportunity\_id 0

opportunity\_name 0

category 0

opportunity\_code 1

cohort\_code 13318

cohort\_start\_ts 13318

cohort\_end\_ts 13323

cohort\_size 13323

apply\_ts 310

status 192

email 275

gender 600

birthdate 606

city 604

state 753

zip 3886

dtype: int64

calculating total zeros

#calculating total zeros

cols = [

    'learner\_id', 'country', 'degree', 'institution', 'major',

    'opportunity\_id', 'opportunity\_name', 'category', 'opportunity\_code',

    'cohort\_code', 'cohort\_start\_ts', 'cohort\_end\_ts', 'cohort\_size',

    'apply\_ts', 'status', 'email', 'gender', 'birthdate', 'city',

    'state', 'zip'

]

(data[cols] == 0).sum()

learner\_id 0

country 0

degree 0

institution 0

major 0

opportunity\_id 0

opportunity\_name 0

category 0

opportunity\_code 0

cohort\_code 0

cohort\_start\_ts 0

cohort\_end\_ts 0

cohort\_size 0

apply\_ts 0

status 0

email 0

gender 0

birthdate 0

city 0

state 0

zip 1807

dtype: Int64

NULL VALUES AFTER COUNTING ZEROS

learner\_id 0

country 223

degree 593

institution 934

major 763

opportunity\_id 0

opportunity\_name 0

category 0

opportunity\_code 1

cohort\_code 13318

cohort\_start\_ts 13318

cohort\_end\_ts 13323

cohort\_size 13323

apply\_ts 310

status 192

email 275

gender 600

birthdate 606

city 604

state 753

zip 5693

dtype: int64

shape after dropping null values

data.shape

(94642, 21) rows and columns

Duplicates removing

data[data.duplicated()]  # Check for any duplicate rows

data.drop\_duplicates(subset=['learner\_id'], inplace=True)

data.shape

(49179, 21) – remaining rows and columns

PRIMARY KEY – LEARNER’ID

FOREIGN KEY – OPPORTUNITIES ID