Step-1: Import packages

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
```

Step-2:Read the data

```
In [5]: file_path=r'C:\Users\omkar\OneDrive\Documents\Gen_AI\Data_files\Visadataset.csv'
        pd.read_csv(file_path)
```

Out[5]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_t
0	EZYV01	Asia	High School	N	
1	EZYV02	Asia	Master's	Y	
2	EZYV03	Asia	Bachelor's	N	
3	EZYV04	Asia	Bachelor's	N	
4	EZYV05	Africa	Master's	Y	
...	...	...	...	...	
25475	EZYV25476	Asia	Bachelor's	Y	
25476	EZYV25477	Asia	High School	Y	
25477	EZYV25478	Asia	Master's	Y	
25478	EZYV25479	Asia	Master's	Y	
25479	EZYV25480	Asia	Bachelor's	Y	

25480 rows × 12 columns

```
In [9]: file_path=r'C:\Users\omkar\OneDrive\Documents\Gen_AI\Data_files\bank.csv'
        pd.read_csv(file_path,sep=';')
```

Out[9]:

	age	job	marital	education	default	balance	housing	loan	contact
0	30	unemployed	married	primary	no	1787	no	no	cellular
1	33	services	married	secondary	no	4789	yes	yes	cellular
2	35	management	single	tertiary	no	1350	yes	no	cellular
3	30	management	married	tertiary	no	1476	yes	yes	unknown
4	59	blue-collar	married	secondary	no	0	yes	no	unknown
...	...	...	...	...	...	...	...	...	...
4516	33	services	married	secondary	no	-333	yes	no	cellular
4517	57	self-employed	married	tertiary	yes	-3313	yes	yes	unknown
4518	57	technician	married	secondary	no	295	no	no	cellular
4519	28	blue-collar	married	secondary	no	1137	no	no	cellular
4520	44	entrepreneur	single	tertiary	no	1136	yes	yes	cellular

4521 rows × 17 columns

In [11]:

```
file_path=r'C:\Users\omkar\OneDrive\Documents\Gen_AI\Data_files\Visadataset.csv'
visa_df=pd.read_csv(file_path)
visa_df
```

Out[11]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_t
0	EZYV01	Asia	High School	N	
1	EZYV02	Asia	Master's	Y	
2	EZYV03	Asia	Bachelor's	N	
3	EZYV04	Asia	Bachelor's	N	
4	EZYV05	Africa	Master's	Y	
...	...	...	...	...	
25475	EZYV25476	Asia	Bachelor's	Y	
25476	EZYV25477	Asia	High School	Y	
25477	EZYV25478	Asia	Master's	Y	
25478	EZYV25479	Asia	Master's	Y	
25479	EZYV25480	Asia	Bachelor's	Y	

25480 rows × 12 columns

In [ ]:

```
- shape

- size
- dtypes

- columns

- drop_duplicates

- isnull

- info
```

In [15]:
```python
visa_df.shape
print("number of rows are:",visa_df.shape[0])
print("number of columns are:",visa_df.shape[1])
```

```
number of rows are: 25480
number of columns are: 12
```

In [17]:
```python
visa_df.size
```

Out[17]:
```
305760
```

In [19]:
```python
25480*12
```

Out[19]:
```
305760
```

In [21]:
```python
visa_df.columns
```

Out[21]:
```
Index(['case_id', 'continent', 'education_of_employee', 'has_job_experience',
       'requires_job_training', 'no_of_employees', 'yr_of_estab',
       'region_of_employment', 'prevailing_wage', 'unit_of_wage',
       'full_time_position', 'case_status'],
      dtype='object')
```

In [23]:
```python
l=['A','B','C']
l.index('C')
```

Out[23]:
```
2
```

In [25]:
```python
visa_df.columns.index('case_status')
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[25], line 1
----> 1 visa_df.columns.index('case_status')

AttributeError: 'Index' object has no attribute 'index'
```

In [27]:
```python
list(visa_df.columns)
```

```
Out[27]: ['case_id',
          'continent',
          'education_of_employee',
          'has_job_experience',
          'requires_job_training',
          'no_of_employees',
          'yr_of_estab',
          'region_of_employment',
          'prevailing_wage',
          'unit_of_wage',
          'full_time_position',
          'case_status']
```

```
In [31]: visa_df.columns.to_list()
```

```
Out[31]: ['case_id',
          'continent',
          'education_of_employee',
          'has_job_experience',
          'requires_job_training',
          'no_of_employees',
          'yr_of_estab',
          'region_of_employment',
          'prevailing_wage',
          'unit_of_wage',
          'full_time_position',
          'case_status']
```

```
In [33]: visa_df.index
```

```
Out[33]: RangeIndex(start=0, stop=25480, step=1)
```

```
In [35]: visa_df.columns
```

```
Out[35]: Index(['case_id', 'continent', 'education_of_employee', 'has_job_experience',
                'requires_job_training', 'no_of_employees', 'yr_of_estab',
                'region_of_employment', 'prevailing_wage', 'unit_of_wage',
                'full_time_position', 'case_status'],
               dtype='object')
```

```
In [39]: visa_df.isnull()
```

Out[39]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_trai
0	False	False	False	False	F
1	False	False	False	False	F
2	False	False	False	False	F
3	False	False	False	False	F
4	False	False	False	False	F
...	...	...	...	...	
25475	False	False	False	False	F
25476	False	False	False	False	F
25477	False	False	False	False	F
25478	False	False	False	False	F
25479	False	False	False	False	F

25480 rows × 12 columns

In [43]:
```
visa_df.shape()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[43], line 1
----> 1 visa_df.shape()

TypeError: 'tuple' object is not callable
```

In [45]:
```
visa_df.isnull().sum()
```

Out[45]:
```
case_id                  0
continent                0
education_of_employee    0
has_job_experience       0
requires_job_training    0
no_of_employees          0
yr_of_estab              0
region_of_employment     0
prevailing_wage          0
unit_of_wage             0
full_time_position       0
case_status              0
dtype: int64
```

In [47]:
```
visa_df.drop_duplicates()
```

Out[47]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_t
0	EZYV01	Asia	High School	N	
1	EZYV02	Asia	Master's	Y	
2	EZYV03	Asia	Bachelor's	N	
3	EZYV04	Asia	Bachelor's	N	
4	EZYV05	Africa	Master's	Y	
...	...	...	...	...	
25475	EZYV25476	Asia	Bachelor's	Y	
25476	EZYV25477	Asia	High School	Y	
25477	EZYV25478	Asia	Master's	Y	
25478	EZYV25479	Asia	Master's	Y	
25479	EZYV25480	Asia	Bachelor's	Y	

25480 rows × 12 columns

In [49]:
```
visa_df1=visa_df.copy()
```

In [51]:
```
visa_df1.shape,visa_df.shape
```

Out[51]:
```
((25480, 12), (25480, 12))
```

In [53]:
```
visa_df.dtypes
```

Out[53]:
```
case_id                   object
continent                 object
education_of_employee     object
has_job_experience        object
requires_job_training     object
no_of_employees            int64
yr_of_estab                int64
region_of_employment      object
prevailing_wage          float64
unit_of_wage              object
full_time_position        object
case_status               object
dtype: object
```

In [55]:
```
type(visa_df.dtypes)
```

Out[55]:
```
pandas.core.series.Series
```

- Series is key-value pair
- Series can form a dictionary when we type cast dict
- Series can also easily convert into dataframe

In [60]:
```python
pd.DataFrame(visa_df.dtypes,
             columns=['Type'])
# TASK : How to make index as column
```

Out[60]:

	Type
case_id	object
continent	object
education_of_employee	object
has_job_experience	object
requires_job_training	object
no_of_employees	int64
yr_of_estab	int64
region_of_employment	object
prevailing_wage	float64
unit_of_wage	object
full_time_position	object
case_status	object

In [72]:
```python
df1=pd.DataFrame(visa_df.dtypes,columns=['Type'])
df1.reset_index(inplace=True)
df1.rename(columns={'index':'Columns'},inplace=True)
df1
```

Out[72]:

	Columns	Type
0	case_id	object
1	continent	object
2	education_of_employee	object
3	has_job_experience	object
4	requires_job_training	object
5	no_of_employees	int64
6	yr_of_estab	int64
7	region_of_employment	object
8	prevailing_wage	float64
9	unit_of_wage	object
10	full_time_position	object
11	case_status	object

In [76]:
```python
visa_df.dtypes.to_dict()
```

```
Out[76]: {'case_id': dtype('O'),
          'continent': dtype('O'),
          'education_of_employee': dtype('O'),
          'has_job_experience': dtype('O'),
          'requires_job_training': dtype('O'),
          'no_of_employees': dtype('int64'),
          'yr_of_estab': dtype('int64'),
          'region_of_employment': dtype('O'),
          'prevailing_wage': dtype('float64'),
          'unit_of_wage': dtype('O'),
          'full_time_position': dtype('O'),
          'case_status': dtype('O')}
```

```
In [82]: list(visa_df.dtypes.to_dict().keys())
```

```
Out[82]: ['case_id',
          'continent',
          'education_of_employee',
          'has_job_experience',
          'requires_job_training',
          'no_of_employees',
          'yr_of_estab',
          'region_of_employment',
          'prevailing_wage',
          'unit_of_wage',
          'full_time_position',
          'case_status']
```

```
In [86]: list(visa_df.dtypes.to_dict().values())
```

```
Out[86]: [dtype('O'),
          dtype('O'),
          dtype('O'),
          dtype('O'),
          dtype('O'),
          dtype('int64'),
          dtype('int64'),
          dtype('O'),
          dtype('float64'),
          dtype('O'),
          dtype('O'),
          dtype('O')]
```

Task-2

- Extract categorical and Numerical Columns in seperate list

```
In [91]: types=visa_df.dtypes.to_dict().values()
         cols=visa_df.dtypes.to_dict().keys()
         for i in types:
             print(i)
```

```
object
object
object
object
object
int64
int64
object
float64
object
object
object
```

In [99]:
```
cat=[]
num=[]
data_types=visa_df.dtypes.to_dict().items()
for i,j in data_types:
    if j=='object':
        cat.append(i)
    else:
        num.append(i)
```

In [103…
```
data_types=visa_df.dtypes.to_dict().items()
cat=[i for i,j in data_types if j=='object']
num=[i for i,j in data_types if j!='object']
```

select_dtypes

In [108…
```
visa_df.select_dtypes(include='object').columns
```

Out[108…
```
Index(['case_id', 'continent', 'education_of_employee', 'has_job_experience',
       'requires_job_training', 'region_of_employment', 'unit_of_wage',
       'full_time_position', 'case_status'],
      dtype='object')
```

In [110…
```
visa_df.select_dtypes(exclude='object').columns
```

Out[110…
```
Index(['no_of_employees', 'yr_of_estab', 'prevailing_wage'], dtype='object')
```

In [ ]:
```

```