In [ ]:
```
import pandas as pd
```

In [ ]:
```
df0 = pd.read_csv('3.csv')
```

In [ ]:
```
df0.head()
```

Out[ ]:

	SKU	CONDITION	SIZE	GENDER	SOLD_AT	SOLD_PRICE	SOURCE	SIZE_VALUE	BRAND	NAME	COLORWAY	COLOR	SILHOUETTE	RETAILPRICE	RELEASEDATE	IS_COLLAB	COLLABORATOR
0	555088-702	is_new	8.5	men	2023-01-25T14:24:20Z	128.0	stockx	8.5	Jordan	Jordan 1 Retro High OG Visionaire	Volt/Black/Sail	green	Air Jordan 1	170.0	2022-06-11	False	NaN
1	EF2829	is_new	7.5	men	2022-03-12T17:39:45Z	340.0	stockx	7.5	adidas	adidas Yeezy Boost 700 V2 Static (2018/2022)	Static/Static/Static	grey	Yeezy Boost 700	300.0	2018-12-29	True	kanye west
2	GX2086	Brand New	11.5	men	2023-07-01T00:00:00Z	99.0	ebay	11.5	adidas	adidas NMD R1 V3 Crystal White Blue Rush	Crystal White/Cloud White/Blue Rush	white	NMD_V3	160.0	2022-09-20	False	NaN
3	DH7863-100	is_new	11	men	2022-05-02T18:12:39Z	190.0	stockx	11.0	Nike	Nike Blazer Low Off-White University Red	White/University Red/Off White	red	Blazer	140.0	2022-04-08	False	NaN
4	CD8180-100	is_new	5.5W	women	2022-04-15T17:49:50Z	150.0	stockx	5.5	Nike	Nike Waffle Racer Off-White White (Women's)	White/Electric Green-Black	green	Waffle Racer	150.0	2019-12-12	False	NaN

In [ ]:
```
df0.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6122327 entries, 0 to 6122326
Data columns (total 17 columns):
 #   Column        Dtype  
---  ------        -----  
 0   SKU           object 
 1   CONDITION     object 
 2   SIZE          object 
 3   GENDER        object 
 4   SOLD_AT       object 
 5   SOLD_PRICE    float64
 6   SOURCE        object 
 7   SIZE_VALUE    float64
 8   BRAND         object 
 9   NAME          object 
 10  COLORWAY      object 
 11  COLOR         object 
 12  SILHOUETTE    object 
 13  RETAILPRICE   float64
 14  RELEASEDATE   object 
 15  IS_COLLAB     object 
 16  COLLABORATOR  object 
dtypes: float64(3), object(14)
memory usage: 794.1+ MB
```

In [ ]:
```
# Calculate the number of null values in each column of DataFrame df0
null_counts = df0.isnull().sum()
print(null_counts)
```

```
SKU              16974
CONDITION            0
SIZE             53096
GENDER           22630
SOLD_AT              0
SOLD_PRICE           0
SOURCE               0
SIZE_VALUE       85522
BRAND            22630
NAME             22630
COLORWAY         25370
COLOR            25370
SILHOUETTE       22686
RETAILPRICE      22630
RELEASEDATE      85234
IS_COLLAB        22619
COLLABORATOR    5260036
dtype: int64
```

In [ ]:
```
# Calculate the percentage of null values for each column
null_percentage = (df0.isnull().sum() / len(df0)) * 100

# Create a DataFrame to display the results
null_percentage_df = pd.DataFrame({
    'Column': null_percentage.index,
    'Null Percentage': null_percentage.values.round(2)
})
```

```
# Print the null percentage DataFrame
print(null_percentage_df)
```

```
          Column  Null Percentage
0            SKU             0.28
1      CONDITION             0.00
2           SIZE             0.87
3         GENDER             0.37
4        SOLD_AT             0.00
5     SOLD_PRICE             0.00
6         SOURCE             0.00
7     SIZE_VALUE             1.40
8          BRAND             0.37
9           NAME             0.37
10      COLORWAY             0.41
11         COLOR             0.41
12    SILHOUETTE             0.37
13   RETAILPRICE             0.37
14   RELEASEDATE             1.39
15     IS_COLLAB             0.37
16  COLLABORATOR            85.92
```

In [ ]:
```
df = df0.dropna(subset=['SKU', 'NAME', 'SIZE_VALUE', 'RELEASEDATE', 'COLORWAY', 'SILHOUETTE'], inplace=True)
```

In [ ]:
```
#Dropping unecessary column
df = df0.drop(columns=['SIZE'])
```

In [ ]:
```
# Removing duplicates
df = df.drop_duplicates()
```

In [ ]:
```
# Try parsing with timezone information
df['SOLD_AT'] = pd.to_datetime(df['SOLD_AT'], format='ISO8601', errors='coerce')

df['RELEASEDATE'] = pd.to_datetime(df['RELEASEDATE'], format='%Y-%m-%d', errors='coerce')
```

In [ ]:
```
# Convert the 'IS_COLLAB' column in the DataFrame 'df' to boolean
df['IS_COLLAB'] = df['IS_COLLAB'].astype(bool)
```

In [ ]:
```
# Update 'COLLABORATOR' to 'None' where 'IS_COLLAB' is False
df.loc[df['IS_COLLAB'] == False, 'COLLABORATOR'] = 'None'
```

In [ ]:
```
# Count null values in 'COLLABORATOR' where 'IS_COLLAB' is True
null_collaborator_count = df.loc[df['IS_COLLAB'] == 1, 'COLLABORATOR'].isnull().sum()

# Print the result
print(f"Number of null values in 'COLLABORATOR' where 'IS_COLLAB' is True: {null_collaborator_count}")
```

```
Number of null values in 'COLLABORATOR' where 'IS_COLLAB' is True: 117180
```

In [ ]:
```
# Identify rows with null "COLLABORATOR"
null_collaborator_rows = df[df['COLLABORATOR'].isnull()]
# Generate distinct values for "COLLABORATOR" based on "SKU"
distinct_collaborator_values = null_collaborator_rows['SKU'].unique()
```

```
# Replace null "COLLABORATOR" values with distinct "SKU" values
df.loc[df['COLLABORATOR'].isnull(), 'COLLABORATOR'] = df.loc[df['COLLABORATOR'].isnull(), 'SKU'].apply(lambda x:distinct_collaborator_values[0])
```

In [ ]:
```
# Count null values in 'COLLABORATOR' where 'IS_COLLAB' is True
null_collaborator_count = df.loc[df['IS_COLLAB'] == 1, 'COLLABORATOR'].isnull().sum()

# Print the result
print(f"Number of null values in 'COLLABORATOR' where 'IS_COLLAB' is True: {null_collaborator_count}")
```

```
Number of null values in 'COLLABORATOR' where 'IS_COLLAB' is True: 0
```

In [ ]:
```
pairs_count = df.groupby(['SIZE_VALUE', 'GENDER']).size().reset_index(name='count')
print(pairs_count)
```

```
     SIZE_VALUE     GENDER  count
0           0.0      child     32
1           0.0     infant   1130
2           0.0        men    199
3           0.0  preschool      6
4           0.0    toddler   1792
..          ...        ...    ...
248        19.5        men      3
249        20.0        men     14
250        27.0      youth      1
251        31.0        men      1
252        44.0        men      1

[253 rows x 3 columns]
```

In [ ]:
```
#removes rows from the DataFrame where the 'SIZE_VALUE' column contains values greater than 16.0.
df.drop(df[df['SIZE_VALUE'] > 16.0].index, inplace=True)
```

In [ ]:
```
# Run the code to change the gender
child_indices = df[(df['SIZE_VALUE'] < 6) & (df['GENDER'] == 'men')].index
df.loc[child_indices, 'GENDER'] = 'child'
```

In [ ]:
```
# Run the code to change the gender
child_indices = df[(df['SIZE_VALUE'] < 6) & (df['GENDER'] == 'unisex')].index
df.loc[child_indices, 'GENDER'] = 'child'
```

In [ ]:
```
# Run the code to change the gender
child_indices = df[(df['SIZE_VALUE'] < 4) & (df['GENDER'] == 'women')].index
df.loc[child_indices, 'GENDER'] = 'child'
```

In [ ]:
```
# Convert 'SIZE_VALUE' to a categorical variable
df['SIZE_VALUE'] = df['SIZE_VALUE'].astype('category')
```

In [ ]:
```
df.isnull().sum()
```

Out[ ]:
```
SKU             0
CONDITION       0
GENDER          0
SOLD_AT         0
SOLD_PRICE      0
SOURCE          0
SIZE_VALUE      0
BRAND           0
NAME            0
COLORWAY        0
COLOR           0
SILHOUETTE      0
RETAILPRICE     0
RELEASEDATE     0
IS_COLLAB       0
COLLABORATOR    0
dtype: int64
```

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 5946878 entries, 0 to 6122326
Data columns (total 16 columns):
 #   Column        Dtype
---  ------        -----
 0   SKU           object
 1   CONDITION     object
 2   GENDER        object
 3   SOLD_AT       datetime64[ns, UTC]
 4   SOLD_PRICE    float64
 5   SOURCE        object
 6   SIZE_VALUE    category
 7   BRAND         object
 8   NAME          object
 9   COLORWAY      object
 10  COLOR         object
 11  SILHOUETTE    object
 12  RETAILPRICE   float64
 13  RELEASEDATE   datetime64[ns]
 14  IS_COLLAB     bool
 15  COLLABORATOR  object
dtypes: bool(1), category(1), datetime64[ns, UTC](1), datetime64[ns](1), float64(2), object(10)
memory usage: 820.9+ MB
```

In [ ]:
```
# extracting cleaned data into csv file
df.to_csv('cleaned_data.csv', index=False)
```