Part 1: Data Cleaning and Preparation (Python)

```python
import pandas as pd
import numpy as np

# Load the data
sales_df = pd.read_csv('sales_data.csv')
employee_df = pd.read_csv('employee_data.csv')
```

Clean Sales Data:

```python
# Convert `SalesAmount` to numeric after removing symbols and commas.
sales_df['SalesAmount'] = sales_df['SalesAmount'].replace('[\$,]', '',
regex=True).astype(float)
sales_df['SalesAmount'] = pd.to_numeric(sales_df['SalesAmount'], errors='coerce')
print(sales_df)
```

```
   TransactionID CustomerID  SalesAmount PurchaseDate  EmployeeID
0         T00001      C0001       2824.0   01-11-2022           1
1         T00002      C0002       1409.0   19-11-2023           2
2         T00003      C0003       5506.0   09-10-2023           3
3         T00004      C0004       5012.0   04-02-2022           3
4         T00005      C0005       4657.0   28-10-2024           7
..           ...        ...          ...          ...         ...
95        T00096      C0096       4593.0   13-08-2022           5
96        T00097      C0097       3266.0   18-09-2023           8
97        T00098      C0098       9348.0   22-05-2022           4
98        T00099      C0099       9085.0   16-11-2022           9
99        T00100      C0100       2489.0   22-06-2024           3

[100 rows x 5 columns]

<>:2: SyntaxWarning: invalid escape sequence '\$'
<>:2: SyntaxWarning: invalid escape sequence '\$'
C:\Users\ziadz\AppData\Local\Temp\ipykernel_26520\679283858.py:2: SyntaxWarning: invalid
escape sequence '\$'
  sales_df['SalesAmount'] = sales_df['SalesAmount'].replace('[\$,]', '',
regex=True).astype(float)
```

```python
# Standardize `PurchaseDate` to pandas `datetime`.
sales_df['PurchaseDate'] = pd.to_datetime(sales_df['PurchaseDate'], errors='coerce',
dayfirst=True)
sales_df
```

```
   TransactionID CustomerID  SalesAmount PurchaseDate  EmployeeID
0         T00001      C0001       2824.0   2022-11-01           1
1         T00002      C0002       1409.0   2023-11-19           2
2         T00003      C0003       5506.0   2023-10-09           3
3         T00004      C0004       5012.0   2022-02-04           3
4         T00005      C0005       4657.0   2024-10-28           7
..           ...        ...          ...          ...         ...
95        T00096      C0096       4593.0   2022-08-13           5
96        T00097      C0097       3266.0   2023-09-18           8
97        T00098      C0098       9348.0   2022-05-22           4
98        T00099      C0099       9085.0   2022-11-16           9
99        T00100      C0100       2489.0   2024-06-22           3
```

```
[100 rows x 5 columns]

# Handle missing or invalid data by replacing them with appropriate defaults (e.g., `NaN`
for missing data).
sales_df.fillna(0 ,inplace=True)
sales_df
sales_df.dtypes

TransactionID             object
CustomerID                object
SalesAmount              float64
PurchaseDate      datetime64[ns]
EmployeeID                 int64
dtype: object
```

Clean Employee Data:

```
employee_df

   EmployeeID             Name  DepartmentID    Salary  SupervisorID
0           1  Cheyenne Padilla             5   $96,438           NaN
1           2    Michael Martin             5  $105,519           6.0
2           3        Tim Wright             4  $103,883           5.0
3           4     Kristy Archer             2  $111,213           1.0
4           5       Robert Rios             4  $145,561           3.0
5           6     Gregory Casey             4   $57,100           3.0
6           7     Douglas Huber             2  $138,259           3.0
7           8    Bobby Browning             1  $135,649           8.0
8           9    Crystal Wilson             1  $134,696           7.0
9          10       Tammy Adams             4   $62,899           2.0

# Convert `Salary` to numeric after removing symbols and commas.
employee_df['Salary'] = employee_df['Salary'].replace('[\$,]', '',
regex=True).astype(float)
employee_df

<>:2: SyntaxWarning: invalid escape sequence '\$'
<>:2: SyntaxWarning: invalid escape sequence '\$'
C:\Users\ziadz\AppData\Local\Temp\ipykernel_26520\3334977909.py:2: SyntaxWarning: invalid
escape sequence '\$'
  employee_df['Salary'] = employee_df['Salary'].replace('[\$,]', '',
regex=True).astype(float)

   EmployeeID             Name  DepartmentID    Salary  SupervisorID
0           1  Cheyenne Padilla             5   96438.0           NaN
1           2    Michael Martin             5  105519.0           6.0
2           3        Tim Wright             4  103883.0           5.0
3           4     Kristy Archer             2  111213.0           1.0
4           5       Robert Rios             4  145561.0           3.0
5           6     Gregory Casey             4   57100.0           3.0
6           7     Douglas Huber             2  138259.0           3.0
7           8    Bobby Browning             1  135649.0           8.0
8           9    Crystal Wilson             1  134696.0           7.0
9          10       Tammy Adams             4   62899.0           2.0

employee_df['Salary'] = pd.to_numeric(employee_df['Salary'], errors='coerce')
employee_df.dtypes
```

```
EmployeeID        int64
Name             object
DepartmentID      int64
Salary          float64
SupervisorID    float64
dtype: object
```

```
# Replace inconsistent `EmployeeID` or `SupervisorID` with clean integers.
employee_df['SupervisorID'] = pd.to_numeric(employee_df['SupervisorID'], errors='coerce')
employee_df
```

```
   EmployeeID              Name  DepartmentID    Salary  SupervisorID
0           1  Cheyenne Padilla             5   96438.0           NaN
1           2    Michael Martin             5  105519.0           6.0
2           3        Tim Wright             4  103883.0           5.0
3           4     Kristy Archer             2  111213.0           1.0
4           5       Robert Rios             4  145561.0           3.0
5           6     Gregory Casey             4   57100.0           3.0
6           7     Douglas Huber             2  138259.0           3.0
7           8    Bobby Browning             1  135649.0           8.0
8           9    Crystal Wilson             1  134696.0           7.0
9          10       Tammy Adams             4   62899.0           2.0
```

```
# Handle any missing `SupervisorID` values by filling them with `NaN`
employee_df['SupervisorID'].replace(np.nan, 0, inplace=True)
```

```
C:\Users\ziadz\AppData\Local\Temp\ipykernel_26520\839830229.py:2: FutureWarning: A value
is trying to be set on a copy of a DataFrame or Series through chained assignment using an
inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the
intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col:
value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the
operation inplace on the original object.


  employee_df['SupervisorID'].replace(np.nan, 0, inplace=True)
```

```
employee_df['SupervisorID'] = employee_df['SupervisorID'].astype('Int64')

employee_df
```

```
   EmployeeID              Name  DepartmentID    Salary  SupervisorID
0           1  Cheyenne Padilla             5   96438.0             0
1           2    Michael Martin             5  105519.0             6
2           3        Tim Wright             4  103883.0             5
3           4     Kristy Archer             2  111213.0             1
4           5       Robert Rios             4  145561.0             3
5           6     Gregory Casey             4   57100.0             3
6           7     Douglas Huber             2  138259.0             3
7           8    Bobby Browning             1  135649.0             8
8           9    Crystal Wilson             1  134696.0             7
9          10       Tammy Adams             4   62899.0             2
```

```
# Save cleaned data
sales_df.to_csv('cleaned_sales.csv', index=False)
employee_df.to_csv('cleaned_employees.csv', index=False)
```