**import** pandas **as** pd

​

*# Load the dataset*

file\_path **=** 'C:/Users/Jepbar/Desktop/SQL project/Nashville Housing.csv'

df **=** pd.read\_csv(file\_path)

​

*# Display first few rows to ensure the data is loaded correctly*

print("Dataset Head:")

print(df.head())

​

*# Check for missing values*

print("\nMissing Values:")

missing\_values **=** df.isnull().sum()

print(missing\_values)

​

*# Percentage of missing values*

print("\nPercentage of Missing Values:")

missing\_percentage **=** (df.isnull().sum() **/** len(df)) **\*** 100

print(missing\_percentage)

Dataset Head:

UniqueID ParcelID LandUse \

0 2045 007 00 0 125.00 SINGLE FAMILY

1 16918 007 00 0 130.00 SINGLE FAMILY

2 54582 007 00 0 138.00 SINGLE FAMILY

3 43070 007 00 0 143.00 SINGLE FAMILY

4 22714 007 00 0 149.00 SINGLE FAMILY

PropertyAddress SaleDate SalePrice LegalReference \

0 1808 FOX CHASE DR, GOODLETTSVILLE 9-Apr-13 240000 20130412-0036474

1 1832 FOX CHASE DR, GOODLETTSVILLE 10-Jun-14 366000 20140619-0053768

2 1864 FOX CHASE DR, GOODLETTSVILLE 26-Sep-16 435000 20160927-0101718

3 1853 FOX CHASE DR, GOODLETTSVILLE 29-Jan-16 255000 20160129-0008913

4 1829 FOX CHASE DR, GOODLETTSVILLE 10-Oct-14 278000 20141015-0095255

SoldAsVacant OwnerName \

0 No FRAZIER, CYRENTHA LYNETTE

1 No BONER, CHARLES & LESLIE

2 No WILSON, JAMES E. & JOANNE

3 No BAKER, JAY K. & SUSAN E.

4 No POST, CHRISTOPHER M. & SAMANTHA C.

OwnerAddress Acreage LandValue BuildingValue \

0 1808 FOX CHASE DR, GOODLETTSVILLE, TN 2.3 50000.0 168200.0

1 1832 FOX CHASE DR, GOODLETTSVILLE, TN 3.5 50000.0 264100.0

2 1864 FOX CHASE DR, GOODLETTSVILLE, TN 2.9 50000.0 216200.0

3 1853 FOX CHASE DR, GOODLETTSVILLE, TN 2.6 50000.0 147300.0

4 1829 FOX CHASE DR, GOODLETTSVILLE, TN 2.0 50000.0 152300.0

TotalValue YearBuilt Bedrooms FullBath HalfBath

0 235700.0 1986.0 3.0 3.0 0.0

1 319000.0 1998.0 3.0 3.0 2.0

2 298000.0 1987.0 4.0 3.0 0.0

3 197300.0 1985.0 3.0 3.0 0.0

4 202300.0 1984.0 4.0 3.0 0.0

Missing Values:

UniqueID 0

ParcelID 0

LandUse 0

PropertyAddress 29

SaleDate 0

SalePrice 0

LegalReference 0

SoldAsVacant 0

OwnerName 31216

OwnerAddress 30462

Acreage 30462

LandValue 30462

BuildingValue 30462

TotalValue 30462

YearBuilt 32314

Bedrooms 32320

FullBath 32202

HalfBath 32333

dtype: int64

Percentage of Missing Values:

UniqueID 0.000000

ParcelID 0.000000

LandUse 0.000000

PropertyAddress 0.051348

SaleDate 0.000000

SalePrice 0.000000

LegalReference 0.000000

SoldAsVacant 0.000000

OwnerName 55.272058

OwnerAddress 53.937001

Acreage 53.937001

LandValue 53.937001

BuildingValue 53.937001

TotalValue 53.937001

YearBuilt 57.216212

Bedrooms 57.226836

FullBath 57.017901

HalfBath 57.249854

dtype: float64

In [21]:

*# Remove rows where 'OwnerName', 'OwnerAddress', or 'PropertyAddress' are missing or labeled as 'Unknown'*

df **=** df[df['OwnerName'] **!=** 'Unknown Owner']

df **=** df[df['OwnerAddress'] **!=** 'Unknown Address']

df **=** df[df['PropertyAddress'] **!=** 'Unknown Address']

​

*# Remove rows where any important numerical columns have missing values*

df **=** df.dropna(subset**=**['Acreage', 'LandValue', 'BuildingValue', 'TotalValue', 'YearBuilt', 'Bedrooms', 'FullBath', 'HalfBath'])

​

*# Re-check the size of the dataset after removing 'Unknown' and nulls*

print(f"New dataset size: {df.shape}")

​

*# Re-check for any remaining missing values*

print(df.isnull().sum())

New dataset size: (25243, 18)

UniqueID 0

ParcelID 0

LandUse 0

PropertyAddress 0

SaleDate 0

SalePrice 0

LegalReference 0

SoldAsVacant 0

OwnerName 0

OwnerAddress 0

Acreage 0

LandValue 0

BuildingValue 0

TotalValue 0

YearBuilt 0

Bedrooms 0

FullBath 0

HalfBath 0

dtype: int64

In [17]:

*# Fill missing categorical values with 'Unknown'*

df['OwnerName'].fillna('Unknown Owner', inplace**=True**)

df['OwnerAddress'].fillna('Unknown Address', inplace**=True**)

df['PropertyAddress'].fillna('Unknown Address', inplace**=True**)

*# Fill missing numerical values with the median of each column*

df['Acreage'].fillna(df['Acreage'].median(), inplace**=True**)

df['LandValue'].fillna(df['LandValue'].median(), inplace**=True**)

df['BuildingValue'].fillna(df['BuildingValue'].median(), inplace**=True**)

df['TotalValue'].fillna(df['TotalValue'].median(), inplace**=True**)

df['YearBuilt'].fillna(df['YearBuilt'].median(), inplace**=True**)

df['Bedrooms'].fillna(df['Bedrooms'].median(), inplace**=True**)

df['FullBath'].fillna(df['FullBath'].median(), inplace**=True**)

df['HalfBath'].fillna(df['HalfBath'].median(), inplace**=True**)

​

*# Re-check for any remaining missing values*

print(df.isnull().sum())

​

UniqueID 0

ParcelID 0

LandUse 0

PropertyAddress 0

SaleDate 0

SalePrice 0

LegalReference 0

SoldAsVacant 0

OwnerName 0

OwnerAddress 0

Acreage 0

LandValue 0

BuildingValue 0

TotalValue 0

YearBuilt 0

Bedrooms 0

FullBath 0

HalfBath 0

dtype: int64

In [22]:

*# Print the number of rows and columns in the dataset*

print(f"Number of rows: {df.shape[0]}")

print(f"Number of columns: {df.shape[1]}")

​

​

Number of rows: 25243

Number of columns: 18

In [ ]:

​

In [34]:

​

​

*# Convert 'SaleDate' to MySQL-compatible date format (YYYY-MM-DD)*

df['SaleDate'] **=** pd.to\_datetime(df['SaleDate'], format**=**'%d-%b-%y', errors**=**'coerce').dt.strftime('%Y-%m-%d')

​

*# Save the updated dataset with the converted date format*

df.to\_csv('Nashville\_Housing\_Converted.csv', index**=False**)

​

print("The 'SaleDate' column has been converted and saved to Nashville\_Housing\_Converted.csv")

​

The 'SaleDate' column has been converted and saved to Nashville\_Housing\_Converted.csv

In [36]:

*# Save the cleaned dataset to a new CSV file*

cleaned\_file\_path **=** 'Nashville\_Housing\_Cleaned\_LAST.csv'

df.to\_csv(cleaned\_file\_path, index**=False**)

​

*# Confirm the file has been saved*

print(f"Cleaned dataset saved to {cleaned\_file\_path}")

​

Cleaned dataset saved to Nashville\_Housing\_Cleaned\_LAST.csv