SQL_Project

February 23, 2024

1 SQL Project - Nashville Housing

- Import data libraries
- Read csv file
- Obtain dataset information
- Remove rows w/ missing values
- Duplicate removal
- Creation of data insertion

[ ]:
```
import pandas as pd
import numpy as np
```

[ ]:
```
df = pd.read_csv('Nashville Housing.csv')
df.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  \
0  1808  FOX CHASE DR, GOODLETTSVILLE      April 9, 2013    240000
1  1832  FOX CHASE DR, GOODLETTSVILLE      June 10, 2014    366000
2  1864 FOX CHASE  DR, GOODLETTSVILLE  September 26, 2016    435000
3  1853  FOX CHASE DR, GOODLETTSVILLE   January 29, 2016    255000
4  1829  FOX CHASE DR, GOODLETTSVILLE   October 10, 2014    278000

     LegalReference SoldAsVacant                          OwnerName  \
0  20130412-0036474           No           FRAZIER, CYRENTHA LYNETTE
1  20140619-0053768           No             BONER, CHARLES & LESLIE
2  20160927-0101718           No           WILSON, JAMES E. & JOANNE
3  20160129-0008913           No            BAKER, JAY K. & SUSAN E.
4  20141015-0095255           No  POST, CHRISTOPHER M. & SAMANTHA C.

                            OwnerAddress  Acreage               TaxDistrict  \
0  1808  FOX CHASE DR, GOODLETTSVILLE, TN      2.3  GENERAL SERVICES DISTRICT
```

```
1  1832  FOX CHASE DR, GOODLETTSVILLE, TN      3.5  GENERAL SERVICES DISTRICT
2  1864  FOX CHASE DR, GOODLETTSVILLE, TN      2.9  GENERAL SERVICES DISTRICT
3  1853  FOX CHASE DR, GOODLETTSVILLE, TN      2.6  GENERAL SERVICES DISTRICT
4  1829  FOX CHASE DR, GOODLETTSVILLE, TN      2.0  GENERAL SERVICES DISTRICT

   LandValue  BuildingValue  TotalValue  YearBuilt  Bedrooms  FullBath  \
0    50000.0       168200.0    235700.0     1986.0       3.0       3.0
1    50000.0       264100.0    319000.0     1998.0       3.0       3.0
2    50000.0       216200.0    298000.0     1987.0       4.0       3.0
3    50000.0       147300.0    197300.0     1985.0       3.0       3.0
4    50000.0       152300.0    202300.0     1984.0       4.0       3.0

   HalfBath
0       0.0
1       2.0
2       0.0
3       0.0
4       0.0
```

2 Data Exploration

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 56477 entries, 0 to 56476
Data columns (total 19 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   UniqueID         56477 non-null  int64
 1   ParcelID         56477 non-null  object
 2   LandUse          56477 non-null  object
 3   PropertyAddress  56448 non-null  object
 4   SaleDate         56477 non-null  object
 5   SalePrice        56477 non-null  object
 6   LegalReference   56477 non-null  object
 7   SoldAsVacant     56477 non-null  object
 8   OwnerName        25261 non-null  object
 9   OwnerAddress     26015 non-null  object
 10  Acreage          26015 non-null  float64
 11  TaxDistrict      26015 non-null  object
 12  LandValue        26015 non-null  float64
 13  BuildingValue    26015 non-null  float64
 14  TotalValue       26015 non-null  float64
 15  YearBuilt        24163 non-null  float64
 16  Bedrooms         24157 non-null  float64
 17  FullBath         24275 non-null  float64
```

```
 18  HalfBath         24144 non-null  float64
dtypes: float64(8), int64(1), object(10)
memory usage: 8.2+ MB
```

[ ]:
```
(56477, 19)
```

3 Data Processing

[ ]:
```
# Remove rows with missing values
df = df.dropna()
```

[ ]:
```
# printing and then dropping any duplicate values
print(df.duplicated().sum())

df.drop_duplicates(inplace=True)
```

```
0
```

4 Creating Injection

- I have done this as I wanted to experiment with data processing techniques in python then transfering the output to an SQL database
- Due to the size of the data in the real world this may not be useful.

[ ]:
```
# Define table name
table_name = 'property_sales'  # Replace 'property_sales' with the actual name␣
 ↪of your table

# Clean up the data and generate SQL INSERT statements
sql_inserts = []
for index, row in df.head(5).iterrows():  # Iterate over the first 5 rows
    # Construct column names and corresponding values dynamically
    columns_values = ", ".join([f"'{str(row[col]).strip().replace("'", "''")}'"␣
 ↪for col in df.columns])
    # Generate the SQL INSERT statement for each row
    sql_insert = f"INSERT INTO {table_name} VALUES ({columns_values});"
    sql_inserts.append(sql_insert)

# Print the first 5 SQL INSERT statements
for sql_insert in sql_inserts:
    print(sql_insert)
```

```
INSERT INTO property_sales VALUES ('2045', '007 00 0 125.00', 'SINGLE FAMILY',
'1808  FOX CHASE DR, GOODLETTSVILLE', 'April 9, 2013', '240000',
'20130412-0036474', 'No', 'FRAZIER, CYRENTHA LYNETTE', '1808  FOX CHASE DR,
GOODLETTSVILLE, TN', '2.3', 'GENERAL SERVICES DISTRICT', '50000.0', '168200.0',
'235700.0', '1986.0', '3.0', '3.0', '0.0');
INSERT INTO property_sales VALUES ('16918', '007 00 0 130.00', 'SINGLE FAMILY',
```

```
'1832  FOX CHASE DR, GOODLETTSVILLE', 'June 10, 2014', '366000',
'20140619-0053768', 'No', 'BONER, CHARLES & LESLIE', '1832  FOX CHASE DR,
GOODLETTSVILLE, TN', '3.5', 'GENERAL SERVICES DISTRICT', '50000.0', '264100.0',
'319000.0', '1998.0', '3.0', '3.0', '2.0');
INSERT INTO property_sales VALUES ('54582', '007 00 0 138.00', 'SINGLE FAMILY',
'1864 FOX CHASE  DR, GOODLETTSVILLE', 'September 26, 2016', '435000',
'20160927-0101718', 'No', 'WILSON, JAMES E. & JOANNE', '1864  FOX CHASE DR,
GOODLETTSVILLE, TN', '2.9', 'GENERAL SERVICES DISTRICT', '50000.0', '216200.0',
'298000.0', '1987.0', '4.0', '3.0', '0.0');
INSERT INTO property_sales VALUES ('43070', '007 00 0 143.00', 'SINGLE FAMILY',
'1853  FOX CHASE DR, GOODLETTSVILLE', 'January 29, 2016', '255000',
'20160129-0008913', 'No', 'BAKER, JAY K. & SUSAN E.', '1853  FOX CHASE DR,
GOODLETTSVILLE, TN', '2.6', 'GENERAL SERVICES DISTRICT', '50000.0', '147300.0',
'197300.0', '1985.0', '3.0', '3.0', '0.0');
INSERT INTO property_sales VALUES ('22714', '007 00 0 149.00', 'SINGLE FAMILY',
'1829  FOX CHASE DR, GOODLETTSVILLE', 'October 10, 2014', '278000',
'20141015-0095255', 'No', 'POST, CHRISTOPHER M. & SAMANTHA C.', '1829  FOX CHASE
DR, GOODLETTSVILLE, TN', '2.0', 'GENERAL SERVICES DISTRICT', '50000.0',
'152300.0', '202300.0', '1984.0', '4.0', '3.0', '0.0');
```

5 Create an output file

- This is useful when dealing with more rows of the data,
- This dataset is huge and could cause issues with my PC,

[ ]:
```python
# Open a file for writing
with open('output.sql', 'w') as file:
    # Write the first 5 SQL INSERT statements to the file
    for sql_insert in sql_inserts[:5]:
        file.write(sql_insert + '\n')
```