# Defining the Question (The Problem Statement).

The data for Nashville housing is in Excel format and needs to be cleaned. In this project, I am going to import the data in SQL and clean it using SQL queries.

# Collecting the Data.

Data was collected from: <https://github.com/AlexTheAnalyst/PortfolioProjects/blob/2dbf63f2f2e8f7c3ff458abc8dc90ddd555f3e38/Nashville%20Housing%20Data%20for%20Data%20Cleaning%20(reuploaded).xlsx>

Snapshot of the data in Excel:

A screenshot of a computer

Description automatically generated

# Cleaning the Data.

We are going to accomplish the following data-cleaning tasks:

1. Standardize the Date Format in SaleDate Field to YYYYMMDD format and remove the time information (which looks redundant)

A screenshot of a computer

Description automatically generated

A screenshot of a computer

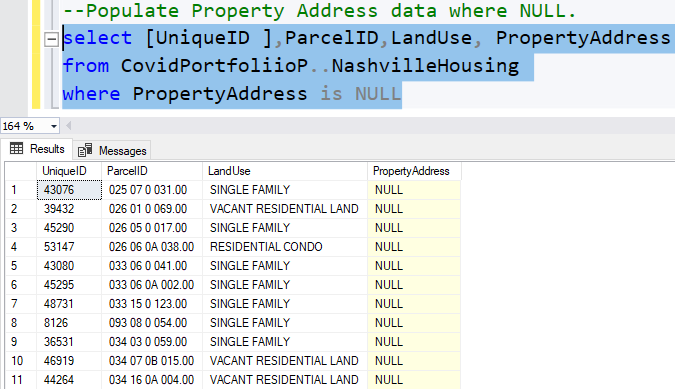
Description automatically generated

The following code adds a new column ConvertedSalesDate as a Date format.

A screenshot of a computer

Description automatically generated

1. Populate the PropertyAddress data where NULL.



The address of the property itself cannot be blank.

We need to figure out why some values in PropertyAddress is NULL and need to populate it by analysing the data.

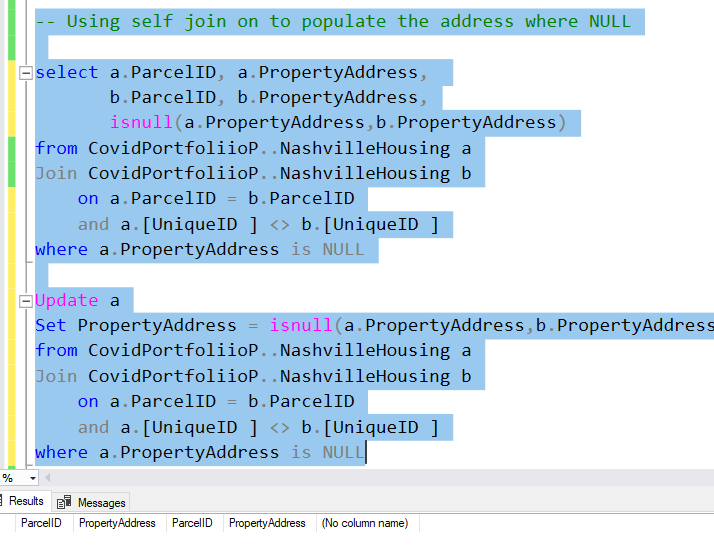
A screenshot of a data

Description automatically generated

By looking at the data closely, I found that some ParcelIDs were duplicate but with unique UniqueIds. And some PropertyAddress were NULL when the ParcelIds were same. Therefore, we can populate the PropertyAddress if NULL from the PropertyAddress if the ParcelIds are same and UniqueIds are different.

This can be a bug in the software. We can also contact the IT support team and figure out if it is a bug so that a ticket can be issued for the Production-Support team.

We need to use Self Join for this.

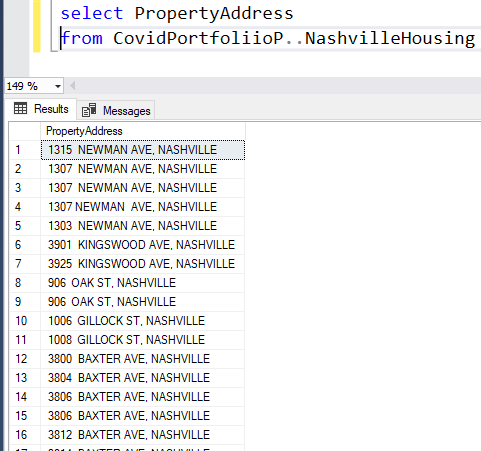


A screenshot of a computer

Description automatically generated

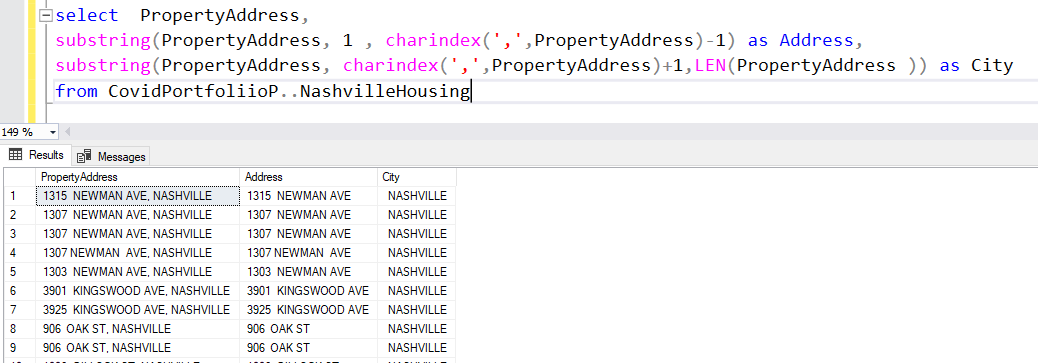
NULL values in the PropertyAddress field has been populated.

1. Breaking out Address into Individual Columns (Address, City, State) from PropertyAddress field.



The column PropertyAddress has the Address and the name of the City separated by a comma (as a delimiter). We need to separate this out into different fields.

Using a select statement before altering/updating the table:



A screenshot of a computer

Description automatically generated

We can see the new columns at the end.

A screenshot of a computer

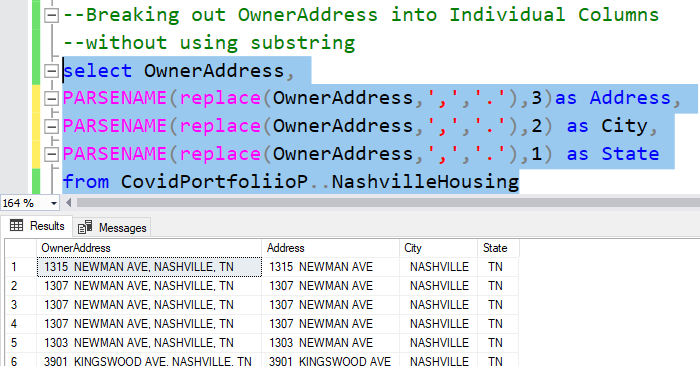
Description automatically generated

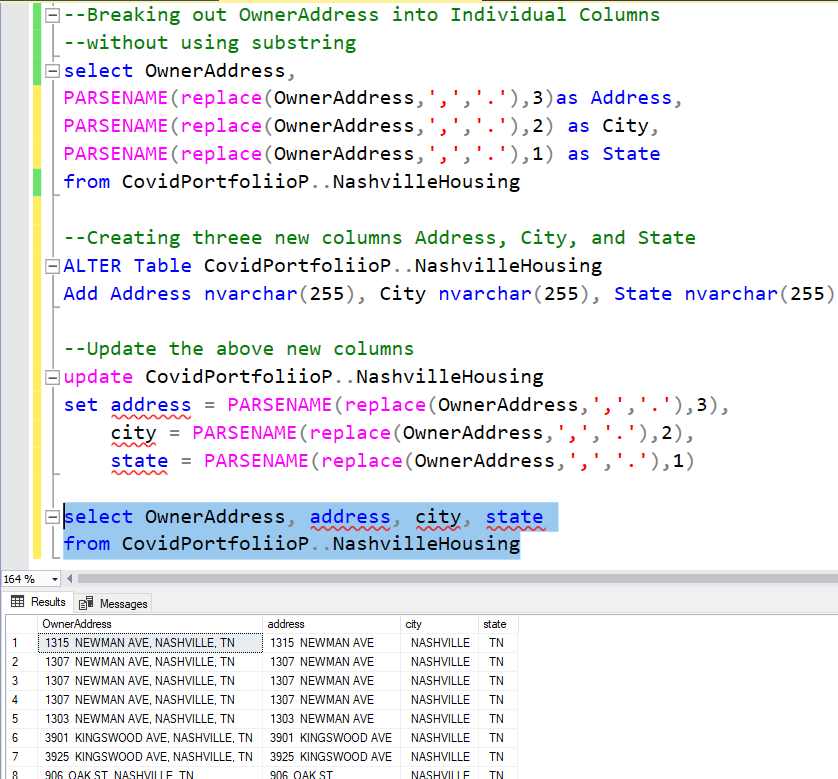
Performing the same task for the OwnerAddress column but without using substring function, but using PARSENAME and REPLACE functions in SQL.

A screenshot of a computer

Description automatically generated

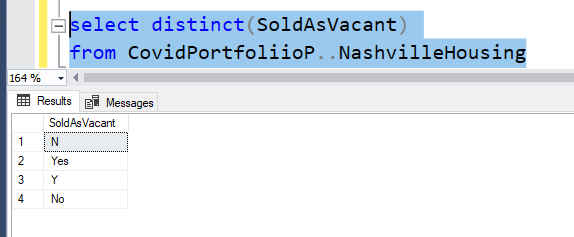
Using a select statement before altering the table:



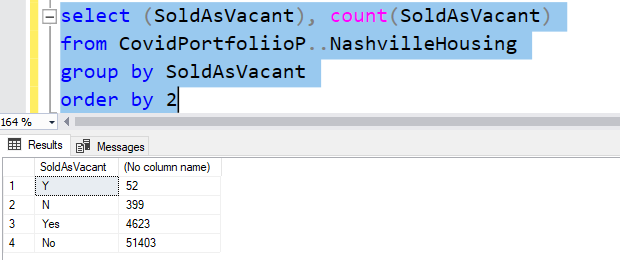


1. Change “Y” and “N” to “Yes” and “No” in the "Sold as Vacant" field.

Column SoldAsVacant has multiple records as “N”, “NO”, “Y”, and “Yes”. We need to change “Yes” and “No” to “Y” and “N” respectively.



But looking at the data, it would require less time to convert “Y” to “Yes” and “N” to “NO” instead.



Checking if the sql query worked:

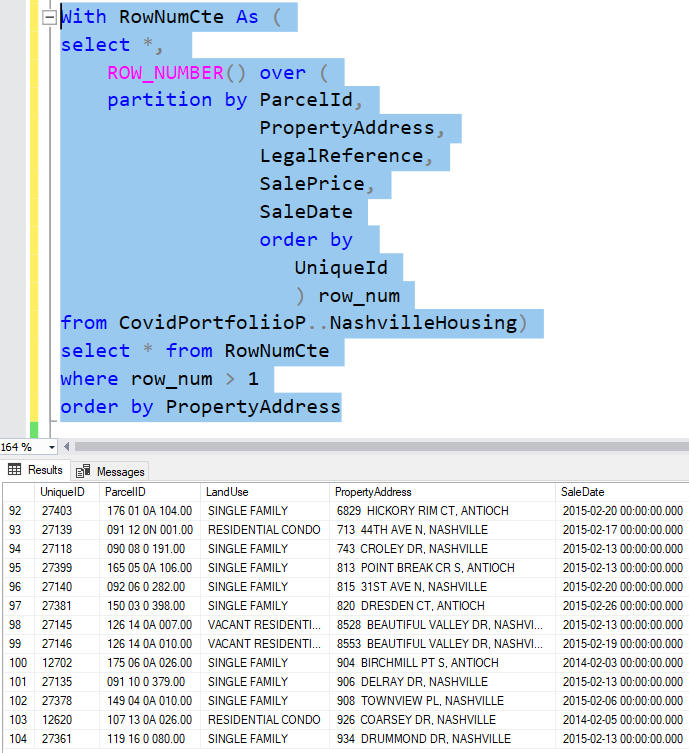
A screenshot of a computer

Description automatically generated

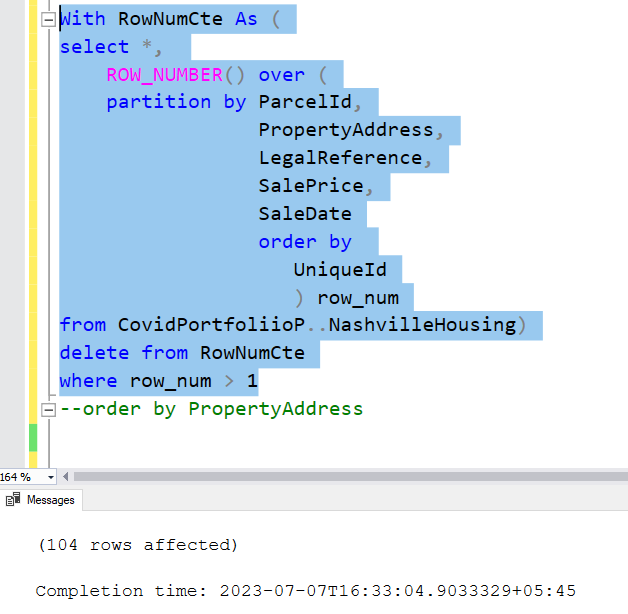
SoldAsVacant column has been updated using the above query.

1. Remove Duplicates, and

Using Window function Row\_number() to find out the duplicate rows:



There are 104 duplicate rows in the table and we need to delete it.



Checking if any duplicate values remain:

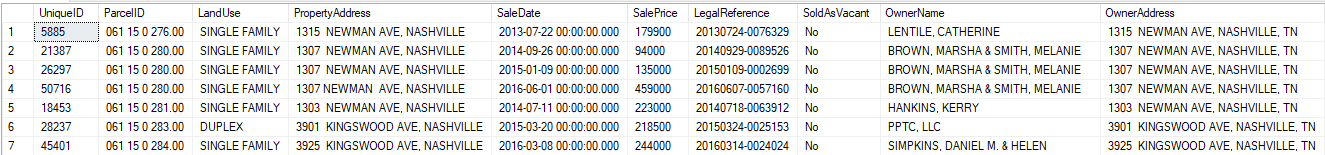
A screenshot of a computer

Description automatically generated

No duplicate values were found.

1. Delete Unused Columns.

We no more require PropertAddress and OwnerAddress columns and can be dropped. However, this is not a good practice to delete raw data in real life.



A screenshot of a computer

Description automatically generated

Checking if the columns exist.

A screenshot of a computer

Description automatically generated

PropertAddress and OwnerAddress columns have been dropped from the table.