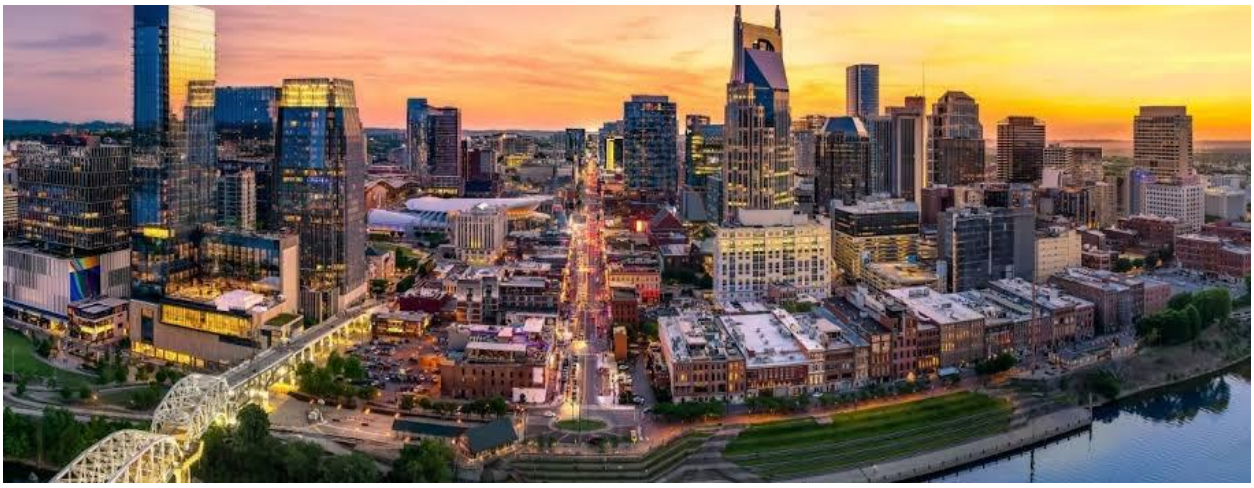
NASHVILLE HOUSING DATA CLEANING USING SQL



In this project we are going to clean the dirty Nashville housing dataset using Microsoft Server Management Studio.

We are going to do the following task:

1. Standardize date format
2. Parsing long formatted address into individual columns (Address, City and State)
3. Populate missing Property address data.
4. Standardize “sold as Vacant” field (from Y/N to Yes and No)
5. Remove duplicate

The Data

Home Value data for the booming Nashville market with 56,000+ rows altogether. The dataset can be found on Kaggle.

*SELECT **
FROM Nashville_housing_data

100 %

Results | Messages

	UniqueID	ParcelID	LandUse	PropertyAddress	SaleDate	SalePrice	LegalReference	SoldAsVacant	OwnerName	OwnerAddress
1	46592	131 01 0D 123.00	RESIDENTIAL CONDO	5025 HILLSBORO PIKE, NASHVILLE	2016-04-15 00:00:00.000	260000	20160418-0037135	No	NULL	NULL
2	23749	131 01 0D 128.00	RESIDENTIAL CONDO	5025 HILLSBORO PIKE, NASHVILLE	2014-11-10 00:00:00.000	135000	20141113-0104831	No	NULL	NULL
3	55774	131 01 0D 129.00	RESIDENTIAL CONDO	5025 HILLSBORO PIKE, NASHVILLE	2016-10-19 00:00:00.000	250000	20161024-0111876	No	NULL	NULL
4	14237	131 01 0D 136.00	RESIDENTIAL CONDO	5025 HILLSBORO PIKE, NASHVILLE	2014-04-15 00:00:00.000	160000	20140416-0031895	No	NULL	NULL
5	6777	131 01 0D 138.00	RESIDENTIAL CONDO	5025 HILLSBORO PIKE, NASHVILLE	2013-08-23 00:00:00.000	167000	20130826-0089613	No	NULL	NULL
6	22478	131 01 0D 141.00	RESIDENTIAL CONDO	5025 HILLSBORO PIKE, NASHVILLE	2014-10-27 00:00:00.000	137500	20141029-0099768	No	NULL	NULL
7	32708	131 01 0D 143.00	RESIDENTIAL CONDO	5025 HILLSBORO PIKE, NASHVILLE	2015-06-01 00:00:00.000	194000	20150603-0051657	No	NULL	NULL
8	23750	131 01 0D 144.00	RESIDENTIAL CONDO	5025 HILLSBORO PIKE, NASHVILLE	2014-11-20 00:00:00.000	195700	20141125-0108419	No	NULL	NULL
9	29219	131 01 0D 145.00	RESIDENTIAL CONDO	5025 HILLSBORO PIKE, NASHVILLE	2015-04-23 00:00:00.000	180000	20150504-0040316	No	NULL	NULL
10	48352	131 01 0D 145.00	RESIDENTIAL CONDO	5025 HILLSBORO PIKE, NASHVILLE	2016-05-23 00:00:00.000	240000	20160531-0054064	No	NULL	NULL

Standardize Date Format

On 'SaleDate' column, we can see that the current format of date is in YYY-MM-DD HH:MM:SS Since the value of HH:MM:SS are all 0, therefore, we will get rid of HH:MM:SS.

SELECT SaleDate
FROM Nashville_housing_data

Output:

100 %

Results | Messages

	SaleDate
1	2016-04-15 00:00:00.000
2	2014-11-10 00:00:00.000
3	2016-10-19 00:00:00.000
4	2014-04-15 00:00:00.000
5	2013-08-23 00:00:00.000
6	2014-10-27 00:00:00.000
7	2015-06-01 00:00:00.000
8	2014-11-20 00:00:00.000
9	2015-04-23 00:00:00.000
10	2016-05-23 00:00:00.000
11	2013-12-18 00:00:00.000

ALTER TABLE Nashville_housing_data
ADD SaleDateConverted DATE

UPDATE Nashville_housing_data
SET SaleDateConverted = CONVERT (DATE, SaleDate)

SELECT SaleDateConverted
FROM Nashville_housing_data

Output:

Results | Messages

	SaleDateConverted
1	2016-04-15
2	2014-11-10
3	2016-10-19
4	2014-04-15
5	2013-08-23
6	2014-10-27
7	2015-06-01
8	2014-11-20
9	2015-04-23
10	2016-05-23
11	2013-12-18

Populate missing Property address data

There are plenty missing values in PropertyAddress.

*SELECT **
FROM Nashville_housing_data
WHERE PropertyAddress IS NULL
ORDER BY ParcelID

Output:

00 %

Results | Messages

	UniqueID	ParcelID	LandUse	PropertyAddress	SaleDate	SalePrice	LegalReference	SoldAsVacant	OwnerName
1	43076	025 07 0 031.00	SINGLE FAMILY	NULL	2016-01-15 00:00:00.000	179900	20160120-0005776	No	COSTNER, FRED & CAROLYN
2	39432	026 01 0 069.00	VACANT RESIDENTIAL LAND	NULL	2015-10-23 00:00:00.000	153000	20151028-0109602	No	SHACKLEFORD, MICHAEL C., JR.
3	45290	026 05 0 017.00	SINGLE FAMILY	NULL	2016-03-29 00:00:00.000	155000	20160330-0029941	No	TRIPP, MARVIN S. & DEBORAH YOUNG
4	53147	026 06 0A 038.00	RESIDENTIAL CONDO	NULL	2016-08-25 00:00:00.000	144900	20160831-0091567	No	NULL
5	43080	033 06 0 041.00	SINGLE FAMILY	NULL	2016-01-04 00:00:00.000	170000	20160107-0001526	No	FRANK, ZACHARY & NIKI
6	45295	033 06 0A 002.00	SINGLE FAMILY	NULL	2016-03-29 00:00:00.000	210000	20160331-0030709	No	NULL
7	48731	033 15 0 123.00	SINGLE FAMILY	NULL	2016-05-05 00:00:00.000	199900	20160506-0045368	No	COLEMAN, AARON A. & CECIL, CORRIE J.
8	36531	034 03 0 059.00	SINGLE FAMILY	NULL	2015-08-13 00:00:00.000	245000	20150819-0083759	No	DILICK, JOHN MARK & ANNETTE A.
9	46919	034 07 0B 015.00	VACANT RESIDENTIAL LAND	NULL	2016-04-27 00:00:00.000	40000	20160304-0020905	Yes	NULL
10	44264	034 16 0A 004.00	VACANT RESIDENTIAL LAND	NULL	2016-02-04 00:00:00.000	130000	20160205-0011327	Yes	NULL

If we look closer, the UniqueID can be in the same ParcelID. Each ParcelID only had one address, so more than one ID could have the same PropertyAddress if they are in the same ParcelID. Therefore, we can use ParcelID as a reference point to populate the missing address in PropertyAddress.

Using Self-Join, we could populate the null property address with a property address that had the same ParcelID.

SELECT a.ParcelID, a.PropertyAddress, b.ParcelID, b.PropertyAddress
FROM Nashville_housing_data AS a
JOIN Nashville_housing_data AS b
ON a.ParcelID = b.ParcelID
AND a.[UniqueID] <> b.[UniqueID]
WHERE a.PropertyAddress IS NULL
ORDER BY a.ParcelID

Output:

100 %

Results | Messages

	ParcelID	PropertyAddress	ParcelID	PropertyAddress
1	025 07 0 031.00	NULL	025 07 0 031.00	410 ROSEHILL CT, GOODLETTSVILLE
2	026 01 0 069.00	NULL	026 01 0 069.00	141 TWO MILE PIKE, GOODLETTSVILLE
3	026 05 0 017.00	NULL	026 05 0 017.00	208 EAST AVE, GOODLETTSVILLE
4	026 06 0A 038.00	NULL	026 06 0A 038.00	109 CANTON CT, GOODLETTSVILLE
5	033 06 0 041.00	NULL	033 06 0 041.00	1129 CAMPBELL RD, GOODLETTSVILLE
6	033 06 0A 002.00	NULL	033 06 0A 002.00	1116 CAMPBELL RD, GOODLETTSVILLE
7	033 15 0 123.00	NULL	033 15 0 123.00	438 W CAMPBELL RD, GOODLETTSVILLE
8	034 03 0 059.00	NULL	034 03 0 059.00	2117 PAULA DR, MADISON
9	034 03 0 059.00	NULL	034 03 0 059.00	2117 PAULA DR, MADISON
10	034 07 0B 015.00	NULL	034 07 0B 015.00	2524 VAL MARIE DR, MADISON
11	034 07 0B 015.00	NULL	034 07 0B 015.00	2524 VAL MARIE DR, MADISON

Now let's actually update the PropertyAddress.

```
UPDATE a
SET PropertyAddress = IS NULL (a.PropertyAddress , b.PropertyAddress)
FROM Nashville_housing_data AS a
JOIN Nashville_housing_data As b
ON a.ParcelID = b.ParcelID
AND a.[UniqueID] <> b.[UniqueID]
WHERE a.PropertyAddress IS NULL
```

After the update, we could check if there are any null rows left and if the updated rows are filled with correct address.

```
SELECT a.ParcelID , a.PropertyAddress , b.ParcelID, b.PropertyAddress
FROM Nashville_housing_data AS a
JOIN Nashville_housing_data AS b
ON a.ParcelID = b.ParcelID
AND a.[UniqueID] <> b.[UniqueID]
WHERE a.PropertyAddress IS NULL
ORDER BY a.ParcelID
```

Output:

100 %

Results | Messages

	ParcelID	PropertyAddress	ParcelID	PropertyAddress
23	025 04 0 126.00	104 ESSEX CT, GOODLETTSVILLE	025 04 0 126.00	104 ESSEX CT, GOODLETTSVILLE
24	025 04 0 126.00	104 ESSEX CT, GOODLETTSVILLE	025 04 0 126.00	104 ESSEX CT, GOODLETTSVILLE
25	025 07 0 008.00	407 ROSEHILL DR, GOODLETTSVILLE	025 07 0 008.00	407 ROSEHILL DR, GOODLETTSVILLE
26	025 07 0 008.00	407 ROSEHILL DR, GOODLETTSVILLE	025 07 0 008.00	407 ROSEHILL DR, GOODLETTSVILLE
27	025 07 0 031.00	410 ROSEHILL CT, GOODLETTSVILLE	025 07 0 031.00	410 ROSEHILL CT, GOODLETTSVILLE
28	025 07 0 031.00	410 ROSEHILL CT, GOODLETTSVILLE	025 07 0 031.00	410 ROSEHILL CT, GOODLETTSVILLE
29	025 08 0 006.00	207 ROSEHILL DR, GOODLETTSVILLE	025 08 0 006.00	207 ROSEHILL DR, GOODLETTSVILLE
30	025 08 0 006.00	207 ROSEHILL DR, GOODLETTSVILLE	025 08 0 006.00	207 ROSEHILL DR, GOODLETTSVILLE
31	025 12 0 029.00	107 SHEVEL DR, GOODLETTSVILLE	025 12 0 029.00	107 SHEVEL DR, GOODLETTSVILLE
32	025 12 0 029.00	107 SHEVEL DR, GOODLETTSVILLE	025 12 0 029.00	107 SHEVEL DR, GOODLETTSVILLE
33	025 12 0 029.00	107 SHEVEL DR, GOODLETTSVILLE	025 12 0 029.00	107 SHEVEL DR, GOODLETTSVILLE

Parsing long formatted address into individual columns (Address, City and State)

If you recall, The PropertyAddress column contains the address and the city the property is located. We need to separate the address and the city into different columns for future analysis.

SELECT PropertyAddress
FROM Nashville_housing_data

Output:

Results | Messages

	PropertyAddress
57	14 LEXINGTON GRN, NASHVILLE
58	17 LEXINGTON GRN, NASHVILLE
59	108 LEXINGTON CT, NASHVILLE
60	4100 BALDWIN ARBOR, NASHVILLE
61	4104 BALDWIN ARBOR, NASHVILLE
62	4120 BALDWIN ARBOR, NASHVILLE
63	3600 COLEWOOD DR, NASHVILLE
64	2241 B CASTLEMAN DR, NASHVILLE
65	2243 B CASTLEMAN DR, NASHVILLE
66	2237 CASTLEMAN DR, NASHVILLE
67	2231 CASTLEMAN DR, NASHVILLE

SELECT
SUBSTRING (PropertyAddress, 1 , CHARINDEX(' , ' , PropertyAddress) - 1),
SUBSTRING (PropertyAddress , CHARINDEX (' , ' , PropertyAddress) + 1 ,
LEN (PropertyAddress))
FROM Nashville_housing_data

Output:

Results | Messages

	(No column name)	(No column name)
57	14 LEXINGTON GRN	NASHVILLE
58	17 LEXINGTON GRN	NASHVILLE
59	108 LEXINGTON CT	NASHVILLE
60	4100 BALDWIN ARBOR	NASHVILLE
61	4104 BALDWIN ARBOR	NASHVILLE
62	4120 BALDWIN ARBOR	NASHVILLE
63	3600 COLEWOOD DR	NASHVILLE
64	2241 B CASTLEMAN DR	NASHVILLE
65	2243 B CASTLEMAN DR	NASHVILLE
66	2237 CASTLEMAN DR	NASHVILLE
67	2231 CASTLEMAN DR	NASHVILLE

```
ALTER TABLE Nashville_housing_data
ADD PropertySplitAddress NVARCHAR (255)

UPDATE Nashville_housing_data
SET PropertySplitAddress  =
SUBSTRING (PropertyAddress , 1 , CHARINDEX  (‘ , ‘ , PropertyAddress)   -  1)

ALTER TABLE Nashville_housing_data
ADD PropertySplitCity  NVARCHAR (255)

UPDATE Nashville_housing_data
SET PropertySplitCity  =
SUBSTRING (PropertyAddress , CHARINDEX ( ‘ , ‘ , PropertyAddress )   + 1 ,
LEN (PropertyAddress))

SELECT *
FROM Nashville_housing_data
```

Output:

100 %

Results | Messages

	srName	OwnerAddress	Acreage	TaxDistrict	LandValue	BuildingValue	TotalValue	YearBuilt	Bedrooms	FullBath	HalfBath	SaleDateConverted	PropertySplitAddress	PropertySplitCity
1	L	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2016-04-15	5025 HILLSBORO PIKE	NASHVILLE
2	L	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2014-11-10	5025 HILLSBORO PIKE	NASHVILLE
3	L	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2016-10-19	5025 HILLSBORO PIKE	NASHVILLE
4	L	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2014-04-15	5025 HILLSBORO PIKE	NASHVILLE
5	L	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2013-08-23	5025 HILLSBORO PIKE	NASHVILLE
6	L	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2014-10-27	5025 HILLSBORO PIKE	NASHVILLE
7	L	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2015-06-01	5025 HILLSBORO PIKE	NASHVILLE
8	L	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2014-11-20	5025 HILLSBORO PIKE	NASHVILLE
9	L	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2015-04-23	5025 HILLSBORO PIKE	NASHVILLE
10	L	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	2016-05-23	5025 HILLSBORO PIKE	NASHVILLE

For the OwnerAddress, it contains Address, City and State in just a single column. We also need to split them to their own columns as well.

```
ALTER TABLE Nashville_housing_data
ADD OwnerSplitAddress  NVARCHAR (255)

UPDATE Nashville_housing_data
SET OwnerSplitAdress  =
PARSENAME (REPLACE (OwnerAddress, ‘ , ‘ ,  ‘ . ‘ ) ,  3 )
```

ALTER TABLE Nashville_housing_data
ADD OwnerSplitCity NVARCHAR (255)

UPDATE Nashville_housing_data
SET OwnerSplitCity =
PARSENAME (REPLACE (OwnersAddress , ‘ , ‘ , ‘ . ‘) , 2)

ALTER TABLE Nahville_housing_data
ADD OwnerSplitState NVARCHAR (255)

UPDATE Nashville_housing_data
SET OwnerSplitCity =
PARSENAME (REPLACE (OwnersAddress , ‘ , ‘ , ‘ . ‘) , 1)

Output:

100 %

Results | Messages

	Value	BuildingValue	TotalValue	YearBuilt	Bedrooms	FullBath	HalfBath	SaleDateConverted	PropertySplitAddress	PropertySplitCity	OwnerSplitAddress	OwnerSplitCity	OwnerSplitState
236	)00	50500	260500	1947	4	2	0	2015-06-30	1904 WARFIELD DR	NASHVILLE	1904 WARFIEL...	NASHVILLE	TN
237	)0	0	20000	NULL	NULL	NULL	NULL	2016-04-27	1909 B WARFIELD DR	NASHVILLE	1909 B WARFIE...	NASHVILLE	TN
238	)00	8800	219900	1948	2	1	0	2016-08-08	1901 WARFIELD DR	NASHVILLE	1901 WARFIEL...	NASHVILLE	TN
239	)00	0	210000	NULL	NULL	NULL	NULL	2016-01-15	1900 KIMBARK DR	NASHVILLE	1900 KIMBARK ...	NASHVILLE	TN
240	)00	150700	362800	1950	2	1	1	2015-01-15	1804 WARFIELD DR	NASHVILLE	1804 WARFIEL...	NASHVILLE	TN
241	)00	159300	369700	1946	2	2	0	2016-09-06	4111 LONE OAK RD	NASHVILLE	4111 LONE OA...	NASHVILLE	TN
242	)00	0	210000	NULL	NULL	NULL	NULL	2016-07-22	1907 KIMBARK DR	NASHVILLE	1907 KIMBARK ...	NASHVILLE	TN
243	)00	0	210000	NULL	NULL	NULL	NULL	2016-07-22	1905 KIMBARK DR	NASHVILLE	1905 KIMBARK ...	NASHVILLE	TN
244	)00	0	210000	NULL	NULL	NULL	NULL	2016-07-22	1901 KIMBARK DR	NASHVILLE	1901 KIMBARK ...	NASHVILLE	TN
245	)00	89300	301600	1948	3	1	0	2013-10-01	1809 WARFIELD DR	NASHVILLE	1809 WARFIEL...	NASHVILLE	TN

Standardize “Sold as Vacant” field (from Y/N to Yes and No)

There are some inconsistencies in the SoldAsVacant column. We could standardize it to only contain ‘Yes’ and ‘No’ categories.

SELECT DISTINCT (SoldAsVacant) ,
COUNT (SoldAsVacant)
FROM Nashville_housing_data
GROUP BY SoldAsVacant
ORDER BY 2

Output:

100 %

Results | Messages

	SoldAsVacant	(No column name)
1	Y	52
2	N	399
3	Yes	4623
4	No	51403

```
UPDATE Nashville_housing_data
SET SoldAsVacant  =
CASE WHEN SoldAsVacant  =   'Y'   THEN   'Yes'
WHEN SoldAsVacant  = 'N'  THEN 'NO'
ELSE SoldAsVacant
END
```

```
SELECT DISTINCT (SoldAsVacant)
COUNT  (SoldAsVacant)
FROM Nashville_housing_data
GROUP BY SoldAsVacant
ORDER BY 2
```

Output:

100 %

Results | Messages

	SoldAsVacant	(No column name)
1	Yes	4675
2	No	51802

Remove duplicate