Nashville Housing Data Cleaning project

Let's look at the data

```sql
Select *
From nashvillehousing
```

Standardize Date Format

```sql
update
  nashvillehousing
set
  saledate = to_char(
    to_date(saledate, 'MM-DD-YYYY'),
    'YYYY-MM-DD'
  )
```

Let's look at the Property Address Data

```sql
Select *
From NashvilleHousing
Where PropertyAddress is null
```

Populate Property Address data

I notice the same ParcelID and PropertyAddress are listed for different UniqueIDs. I want to it to find a ParcelID with a null PropertyAddress. Then populate the PropertyAddress from a different UniqueID with a matching ParcelID.

```
SELECT
  nx.parcelid,
  nx.propertyaddress,
  ny.parcelid,
  ny.propertyaddress,
  coalesce(nx.propertyaddress, ny.propertyaddres
FROM
  Nashvillehousing nx
  join Nashvillehousing ny on nx.parcelid = ny.p
  and nx.uniqueid <> ny.uniqueid
where
  nx.propertyaddress is null



update
  nashvillehousing
set
  propertyaddress = coalesce(
    nx.propertyaddress, ny.propertyaddress
  )
FROM
  Nashvillehousing nx
  join Nashvillehousing ny on nx.parcelid = ny.p
  and nx.uniqueid <> ny.uniqueid
where
  nx.propertyaddress is null
  and nashvillehousing.propertyaddress is null
```

Breaking out Address into Individual Columns (Address, City, State)

Let's look at the PropertyAddress

```
alter table nashvillehousing
add column PropertySplitAddress varchar(255) nul

alter table nashvillehousing
add column PropertySplitcity varchar(255) null;

update nashvillehousing
set propertysplitaddress = split_part(propertyad
```

```
update nashvillehousing
set propertysplitcity = split_part(propertyaddre
```

Let's look at the OwnerAddress

```
alter table nashvillehousing
add column ownerSplitAddress varchar(255) null;

alter table nashvillehousing
add column ownerSplitcity varchar(255) null;

alter table nashvillehousing
add column ownerSplitstate varchar(255) null;



update nashvillehousing
set ownersplitaddress = split_part(owneraddress,


update nashvillehousing
set ownersplitcity = split_part(owneraddress,','


update nashvillehousing
set ownersplitstate = split_part(owneraddress,',
```

Change Y and N to Yes and No in "Sold as Vacant" field

Let's look at the SoldAsVacant column

```
select soldasvacant, count(soldasvacant)
from nashvillehousing
group by soldasvacant



select case
           when soldsvacant = 'Y' THEN 'Yes'
           when soldsvacant = 'N'THEN 'No'
from nashvilehousing
```

```
update nashvillehousing
set soldasvacant = case
           when soldasvacant = 'Y' THEN 'Yes'
           when soldasvacant = 'N'THEN 'No'
           else soldasvacant end
```

Remove Duplicates

```
select *, row_number()
 over (partition by parcelid,
                    propertyaddress,
                    saledate,
                    legalreference,
                    saleprice
 order by uniqueid) as row_num
from nashvillehousing
```

In the column row_num, I can identify the 2's. Upon investigation, I see the 2 rows have all the same data but different UniqueId's

```
delete from nashvillehousing
where uniqueid in
(select uniqueid from (select *, row_number() ov
(partition by parcelid,
              propertyaddress,
              saledate,
              legalreference,
              saleprice
order by uniqueid) as row_num
from nashvillehousing) x
where x.row_num > 1);
```

Delete Unused Columns

```
ALTER TABLE NashvilleHousing
DROP COLUMN OwnerAddress ,
DROP COLUMN PropertyAddress ,
DROP COLUMN TaxDistrict;
```

These addresses have been split, so delete the old ones.

You also can delete any other unwanted columns, but I'm leaving the rest as is