# Data Cleaning in SQL - Queries

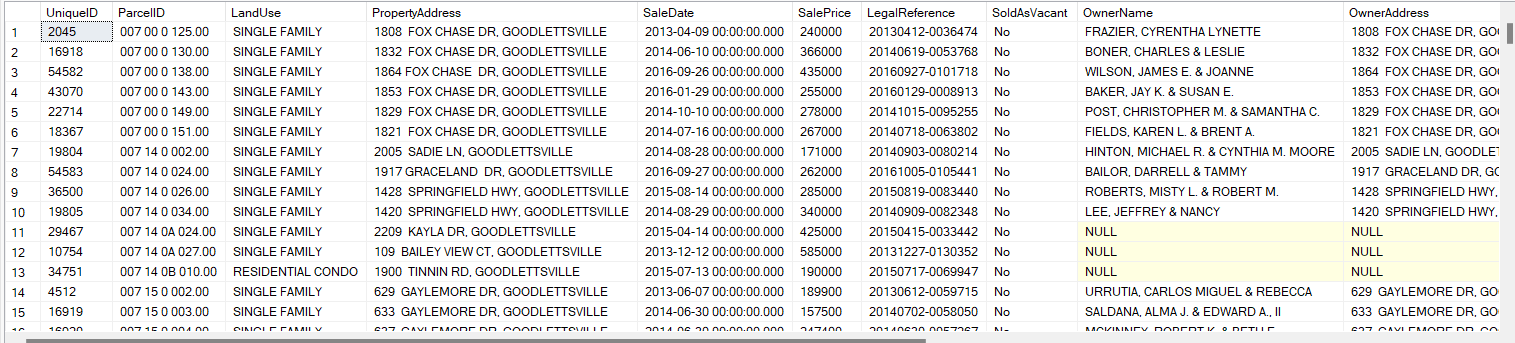
/\*

Cleaning data in SQL

\*/

select \*

from PortfolioProject..Nashvillehousing



Changing the date format

select SaleDateconverted, CONVERT(date,SaleDate) as newsaledate

from PortfolioProject..Nashvillehousing

update Nashvillehousing

set SaleDate = CONVERT(date,SaleDate)

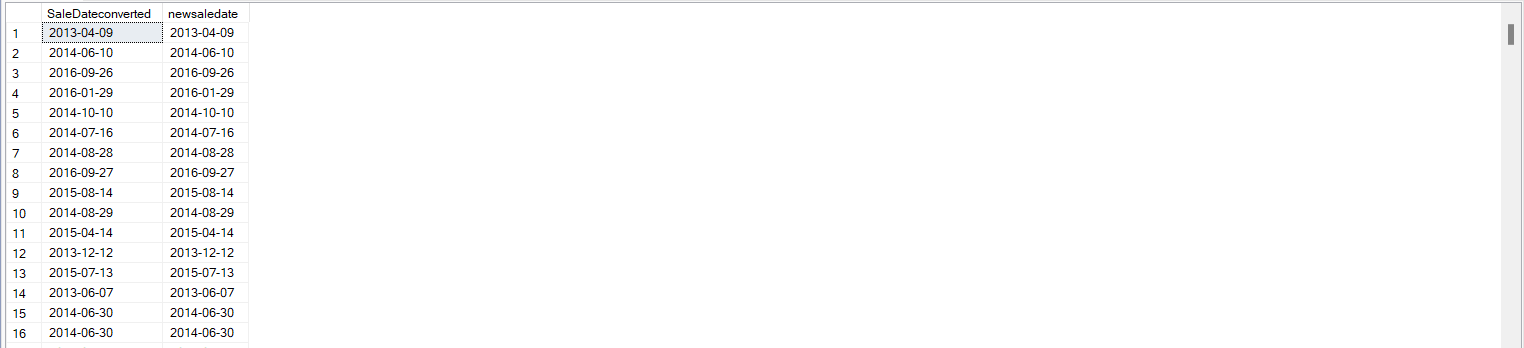
--another method to convert into date format

alter table nashvillehousing

add saledateconverted date;

update Nashvillehousing

set saledateconverted = CONVERT(date,saledate)



Populating the property address

/\* inserting the address while matching it with parcelID, if multiple parcelID

has same addess than replace null with the address\*/

select PropertyAddress

from PortfolioProject..Nashvillehousing

--where PropertyAddress is null

order by ParcelID

/\*Unique id is used to match between two columns- Parcelid and propertyaddress\*/

select a.ParcelID, a.PropertyAddress, b.ParcelID, b.PropertyAddress --, ISNULL(a.PropertyAddress, b.PropertyAddress)

from PortfolioProject..Nashvillehousing a

join PortfolioProject..Nashvillehousing b

on a.ParcelID = b.ParcelID

and a.[UniqueID ]<>b.[UniqueID ]

--where a.PropertyAddress is null /\*after updating the table remove the dashes and run the query again to verify\*/



-- Updating the table with the new column

update a

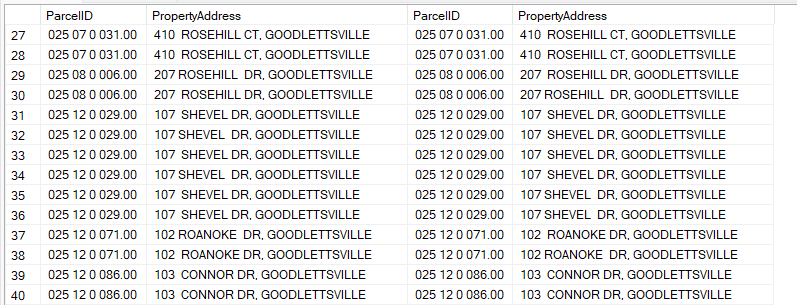
set PropertyAddress = ISNULL(a.PropertyAddress, b.PropertyAddress)

from PortfolioProject..Nashvillehousing a

join PortfolioProject..Nashvillehousing b

on a.ParcelID = b.ParcelID

and a.[UniqueID ]<>b.[UniqueID ]

where a.PropertyAddress is null 

Breaking the address column into individual columns

/\*

in the first substring looks in propertyaddress starts from the first value until comma - 1 to go back 1 place from comma

in the second looks in the propertyaddress strarts from comma till the lenght of the propertyadress \*/

select

SUBSTRING(propertyaddress, 1,CHARINDEX(',',PropertyAddress) -1) as address,

Substring(propertyaddress, CHARINDEX(',',propertyaddress)+1, LEN(propertyaddress)) as address

from PortfolioProject..Nashvillehousing

/\* below alter query is used only to drop the columns\*/

alter table nashvillehousing

drop column propertynewaddress

alter table nashvillehousing

drop column propertynewcity

/\*updating the table by adding new columns for address\*/

alter table nashvillehousing

add propertynewaddress nvarchar(255);

update Nashvillehousing

set propertynewaddress = SUBSTRING(propertyaddress, 1,CHARINDEX(',',PropertyAddress) -1)

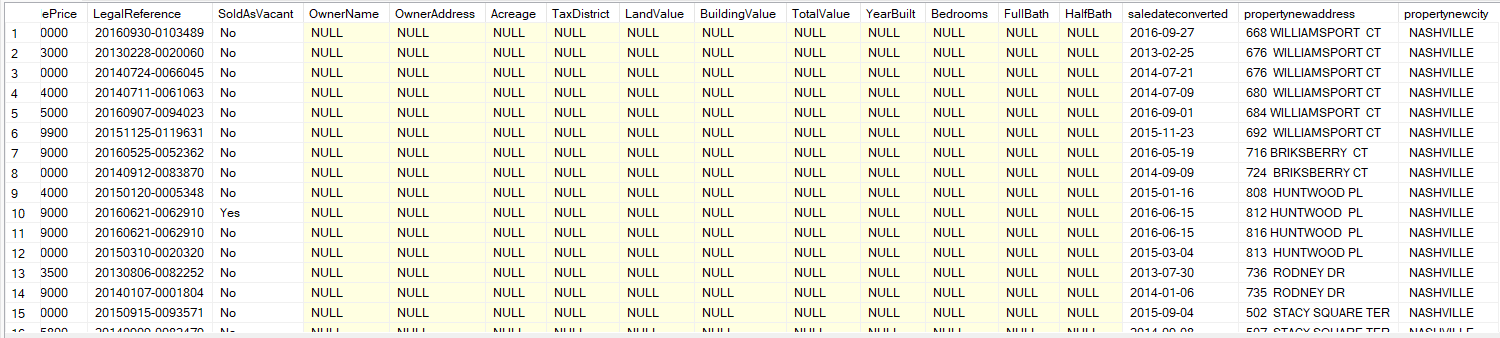
alter table nashvillehousing

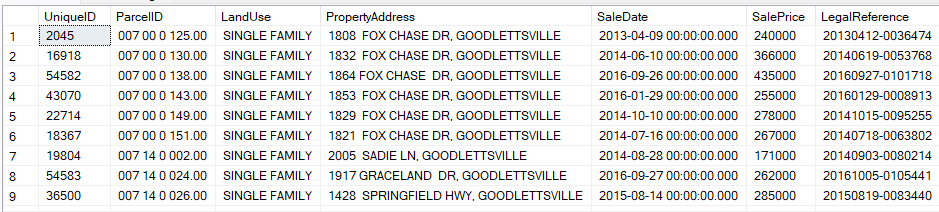
add propertynewcity nvarchar(255);

update Nashvillehousing

set propertynewcity = SUBSTRING(propertyaddress, CHARINDEX(',',propertyaddress)+1, LEN(propertyaddress))

before updating



After updating

Splitting the Owner Address

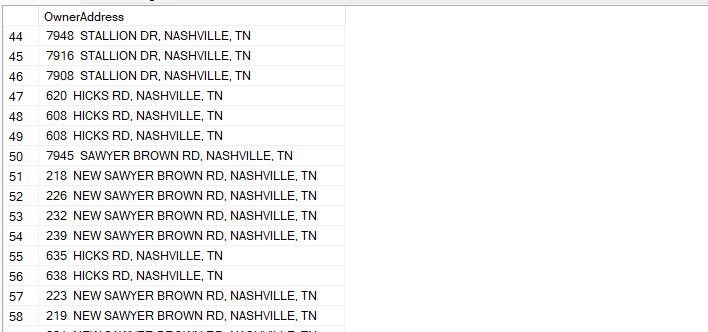
select PARSENAME(replace(OwnerAddress,',','.'),3),

PARSENAME(replace(OwnerAddress,',','.'),2),

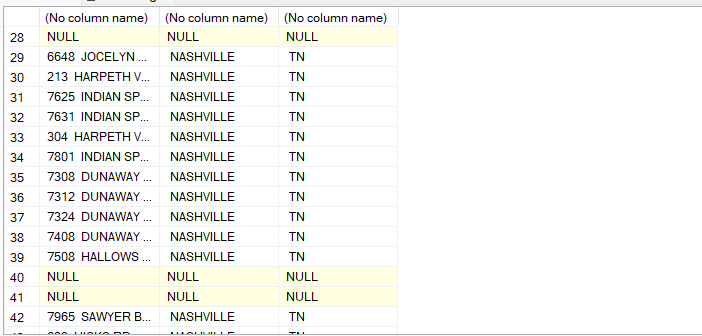
PARSENAME(replace(OwnerAddress,',','.'),1)

from PortfolioProject..Nashvillehousing

Before Splitting



After Splitting



Updating the table by creating new columns for above query

Alter table nashvillehousing

add ownersplitaddress nvarchar(255);

update Nashvillehousing

set ownersplitaddress = PARSENAME(replace(OwnerAddress,',','.'),3)

alter table nashvillehousing

add ownersplitcity nvarchar(255);

update Nashvillehousing

set ownersplitcity = PARSENAME(replace(OwnerAddress,',','.'),2)

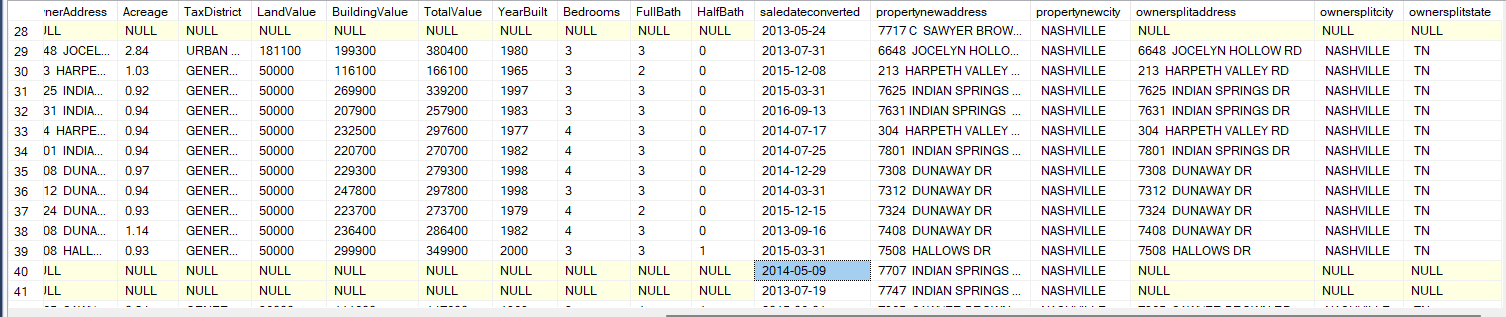
alter table nashvillehousing

add ownersplitstate nvarchar(255);

update Nashvillehousing

set ownersplitstate = PARSENAME(replace(OwnerAddress,',','.'),1)

Updated Table



To Check the data

select Distinct(SoldAsVacant), COUNT(SoldAsVacant)

from PortfolioProject..Nashvillehousing

group by SoldAsVacant

order by 2

select Distinct(SoldAsVacant), COUNT(SoldAsVacant)

from PortfolioProject..Nashvillehousing

group by SoldAsVacant

order by 2

select SoldAsVacant,

case when SoldAsVacant = 'Y' then 'Yes'

when SoldAsVacant = 'N' then 'No'

else SoldAsVacant

end

from PortfolioProject..Nashvillehousing

/\*Upadating the table\*/

update Nashvillehousing

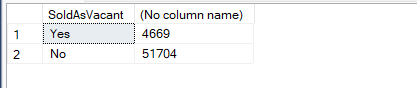
set SoldAsVacant = case when SoldAsVacant = 'Y' then 'Yes'

when SoldAsVacant = 'N' then 'No'

else SoldAsVacant

end

After Updating



Removing Duplicates Using CTE(Common Table Expression)

To check for duplicates using row number (if the row number has the value 2 than there is a duplicate data)

with rownumcte as(

select \*,

row\_number() over(

partition by ParcelID,

PropertyAddress,

SaleDate,

SalePrice,

LegalReference

order by

UniqueID

)row\_num

from PortfolioProject..Nashvillehousing

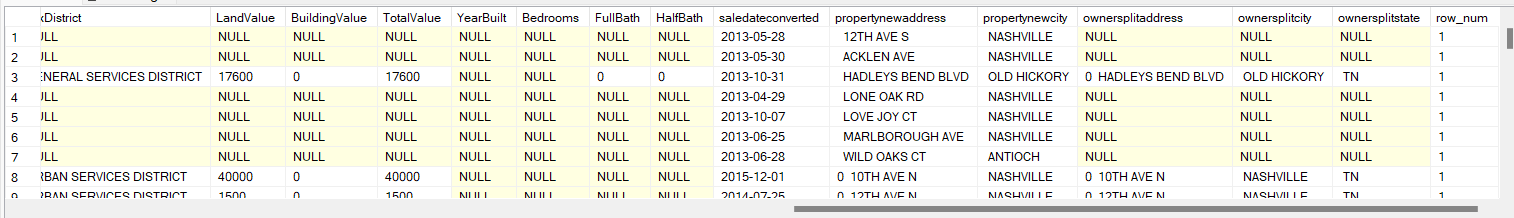
)

select \*

from rownumcte

where row\_num >1

order by PropertyAddress



Deleting the duplicate data from the table

with rownumcte as(

select \*,

row\_number() over(

partition by ParcelID,

PropertyAddress,

SaleDate,

SalePrice,

LegalReference

order by

UniqueID

)row\_num

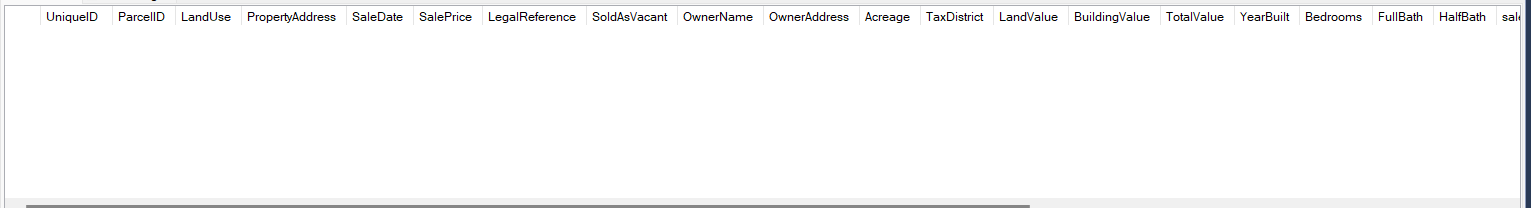
from PortfolioProject..Nashvillehousing

)

Delete

from rownumcte

where row\_num >1

After running the delete query   
 

Deleting the columns from the table

/\*Deleting Unused Columns\*/

select \*

from PortfolioProject..Nashvillehousing

--where TaxDistrict is null

alter table Portfolioproject..nashvillehousing

drop column PropertyAddress, SaleDate, OwnerAddress, TaxDistrict