SQL Data Cleaning Project

Data cleaning in SQL is a crucial step in the data preparation process. It involves identifying and rectifying errors, inconsistencies, and inaccuracies in a dataset to ensure that the data is accurate, reliable, and suitable for analysis.

-- Standardizing Date Format

Select SaleDate, CONVERT(Date, SaleDate) as AfterSaleDateConvert

from PortfolioProject..NashvilleHousing

A screenshot of a computer

Description automatically generated

/\* Query does not update the column (Problem with Dataset) \*/

Update NashvilleHousing

Set SaleDate = CONVERT(Date, SaleDate)

/\* Add a new column with Converted Date \*/

Alter Table NashvilleHousing

Add SaleDateConverted Date;

Update NashvilleHousing

Set SaleDateConverted = CONVERT(Date, SaleDate)

Select SaleDateConverted, CONVERT(Date, SaleDate) as AfterSaleDateConvert

from PortfolioProject..NashvilleHousing

A screenshot of a computer

Description automatically generated

Assigning the Property Address to multiple ParcelIDs sharing the same Property Address. If the ParcelID remains the same, but only the first row of data contains a Property Address, we will fill in the Property Address using the one directly above it since the ParcelID is identicalA blue and white box with black letters

Description automatically generated

Select a.ParcelID, a.PropertyAddress, b.ParcelID, b.PropertyAddress, ISNULL(a.PropertyAddress, b.PropertyAddress)

From PortfolioProject..NashvilleHousing as a

Join PortfolioProject..NashvilleHousing as b

on a.ParcelID = b.ParcelID

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

Where a.PropertyAddress is null

Update a

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

From PortfolioProject..NashvilleHousing as a

Join PortfolioProject..NashvilleHousing as b

on a.ParcelID = b.ParcelID

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

Where a.PropertyAddress is null

Data before update:

A screenshot of a computer

Description automatically generated

After Updating the dataset:

A close up of a logo

Description automatically generated

Splitting the Property Addresses by it’s Address, City and State would help the user identify information easily.

Select PropertyAddress

From PortfolioProject..NashvilleHousing

Data before update:

A close up of a name

Description automatically generated

Select

SUBSTRING(PropertyAddress, 1, CHARINDEX(',', PropertyAddress)-1) as Address,

SUBSTRING(PropertyAddress, CHARINDEX(',', PropertyAddress)+1, LEN(PropertyAddress)) as Address

From PortfolioProject..NashvilleHousing

Data after update:

A close-up of a list of address

Description automatically generated

A more efficient method to identify the Property Address Information

Select

PARSENAME(REPLACE(OwnerAddress, ',', '.'), 3) as OwnerSplitAddress,

PARSENAME(REPLACE(OwnerAddress, ',', '.'), 2) as OwnerSplitCity,

PARSENAME(REPLACE(OwnerAddress, ',', '.'), 1) as OwnerSplitState

From PortfolioProject..NashvilleHousing

A screenshot of a computer

Description automatically generated

Standardizing fields of information. In this case, we are changing fields of information consisting of the letters ‘Y’ and ‘N’ to “Yes” and “No” respectively.

A screenshot of a computer

Description automatically generated

Highlighted portion shows the before and after changes has been made for testing purposes:

Select SoldAsVacant

, CASE When SoldAsVacant = 'Y' Then 'Yes'

When SoldAsVacant = 'N' Then 'No'

ELSE SoldAsVacant

END

From PortfolioProject..NashvilleHousing

A screenshot of a computer

Description automatically generated

Update NashvilleHousing

SET SoldAsVacant = CASE When SoldAsVacant = 'Y' Then 'Yes'

When SoldAsVacant = 'N' Then 'No'

ELSE SoldAsVacant

END

A close up of numbers

Description automatically generated

Removing Duplicated Data

/\* Identifying Duplicates \*/

WITH RowNumCTE AS(

Select \*,

ROW\_NUMBER() OVER (

PARTITION BY ParcelID,

PropertyAddress,

SalePrice,

SaleDate,

LegalReference

ORDER BY

UniqueID

) row\_num

From PortfolioProject..NashvilleHousing

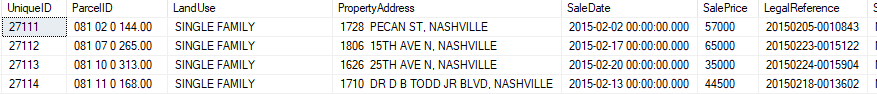
)

Select \*

from RowNumCTE

where row\_num > 1

Data before Removal



WITH RowNumCTE AS(

Select \*,

ROW\_NUMBER() OVER (

PARTITION BY ParcelID,

PropertyAddress,

SalePrice,

SaleDate,

LegalReference

ORDER BY

UniqueID

) row\_num

From PortfolioProject..NashvilleHousing

)

DELETE

from RowNumCTE

where row\_num > 1

Data After Removal

A white background with black text

Description automatically generated

-- Deleting Unused Columns

Select \*

From PortfolioProject..NashvilleHousing

Alter Table PortfolioProject..NashvilleHousing

DROP COLUMN OwnerAddress, TaxDistrict, PropertyAddress, SaleDate

A table with numbers and numbers

Description automatically generated