SQL Data Cleaning Project

Nashville Housing Dataset

Project Overview

This project involves cleaning and preparing the Nashville Housing dataset using SQL. The goal is to ensure the dataset is clean, consistent, and ready for analysis.

Dataset Description

- The dataset contains information on housing properties in Nashville, including:
- Parcel ID, Property Address, Sale Date, and Sale Price
- Land Use, Acreage, Tax District, and Property Value
- Number of Bedrooms, Bathrooms, and Year Built

Data Cleaning Steps

Key cleaning operations performed:

- Handled missing values and null entries
- Removed duplicate records
- Standardized date formats
- Ensured consistency in text fields (e.g., casing, spelling)
- Corrected incorrect or anomalous values

DATA CLEANING USING SQL

Overview of data

- Select count(*) from nashville_housing;

- SELECT * FROM Nashville_housing;

Standardize Data Format

- Select saleprice From nashville_housing;
- SELECT CAST(CAST(SalePrice AS NUMERIC) AS INTEGER) From Nashville_Housing;

- ALTER TABLE Nashville_Housing

  Add SalePriceConverted INTEGER;

- UPDATE Nashville_Housing

  SET SalePriceConverted = SalePrice::NUMERIC::INTEGER;

Populate Property Address data

```
Select *
From Nashville_Housing
-- Where propertyaddress is null
order by parcelid;
```

```
Select a.parcelid, a.propertyaddress, b.parcelid, b.propertyaddress,
COALESCE(a.propertyaddress, b.propertyaddress)
From Nashville_Housing as a
join Nashville_Housing as b
    on a.parcelid = b.parcelid
    AND a.UniqueID <> b.UniqueID
Where a.propertyaddress is null;

update Nashville_HousingSET propertyaddress =
COALESCE(a.propertyaddress, b.propertyaddress)From Nashville_Housing as a
        join Nashville_Housing as b
        on a.parcelid = b.parcelidAND a.UniqueID <> b.UniqueIDWhere
a.propertyaddress is null;
```

Breaking out Address into Individual columns

- Select propertyaddress

  From Nashville_Housing;

- SELECT propertyaddress,

  SUBSTRING(propertyaddress FROM 1 FOR POSITION(',' IN propertyaddress) - 1) AS Address,

  SUBSTRING(propertyaddress

  FROM POSITION(',' IN propertyaddress) + 1) AS Address_RestFROM

  Nashville_Housing;

- ALTER TABLE Nashville_Housing
  ADD ProperySplitAddress varchar(255);

- UPDATE Nashville_Housing
  SET ProperySplitAddress = SUBSTRING(propertyaddress FROM 1 FOR
  POSITION(',' IN propertyaddress) - 1)

- ALTER TABLE Nashville_Housing
  ADD PropertySplitCity varchar(255);

- UPDATE Nashville_Housing
  SET PropertySplitCity = SUBSTRING(propertyaddress FROM
  POSITION(',' IN propertyaddress) + 1)

- Select * from Nashville_Housing

- Select owneraddress from Nashville_Housing;

- SELECT SPLIT_PART(OwnerAddress, ',', 1) AS Part1,
  SPLIT_PART(OwnerAddress, ',', 2) AS Part2,
  SPLIT_PART(OwnerAddress, ',', 3) AS Part3
  FROM Nashville_Housing;

- ALTER TABLE Nashville_Housing
  ADD OwnerSplitAddress varchar(255);

- UPDATE Nashville_Housing
  SET OwnerSplitAddress = SPLIT_PART(OwnerAddress, ',', 3)

- ALTER TABLE Nashville_Housing
  ADD OwnerSplitCity varchar(255);

- UPDATE Nashville_Housing
  SET OwnerSplitCity = SPLIT_PART(OwnerAddress, ',', 2)

- ALTER TABLE Nashville_Housing
  ADD OwnerSplitState varchar(255);

- UPDATE Nashville_Housing
  SET OwnerSplitState = SPLIT_PART(OwnerAddress, ',', 1)

- Select * From Nashville_Housing;

Change Y and N to Yes and No

```
Select Distinct(SoldasVacant), Count(SoldAsVacant)
From Nashville_Housing
Group by SoldAsVacant
Order by 2;
```

Change Y and N to Yes and No

```
Select SoldAsVacant,
        CASE When SoldAsVacant = 'Y' Then 'Yes‘
        When SoldAsVacant = 'N' Then 'No‘
        ELSE SoldAsVacant END
From Nashville_Housing;


Update Nashville_Housing
    SET SoldAsVacant = CASE
    When SoldAsVacant = 'Y' Then 'Yes‘
    When SoldAsVacant = 'N' Then 'No‘
    ELSE SoldAsVacant END
```

Remove Duplicates

```
With RowNumCTE AS(
        Select *,
        Row_NUMBER() OVER(
        PARTITION BY Parcelid, PropertyAddress, SalePrice,
        SaleDate, LegalReference
        Order By Uniqueid) as row_num
        From Nashville_Housing
        -- order by Parcelid)
Select *From RowNumCTE
Where row_num >1
Order by PropertyAddress
```

Delete Unused Columns

```
Select *From Nashville_Housing


ALTER TABLE Nashville_Housing
DROP COLUMN OwnerAddress,
DROP COLUMN TaxDistrict,
DROP COLUMN PropertyAddress,
DROP COLUMN SaleDate;
```

Thank You