SQL Data Cleaning Project: Nashville Housing Data

Project Overview

This project focuses on **cleaning, standardizing, and transforming a raw Nashville Housing dataset** using **SQL (MYSQL WORKBENCH)**. The goal was to convert messy, inconsistent, and incomplete data into a clean, structured format suitable for analysis and reporting.

The script demonstrates several essential data cleaning techniques that are fundamental in any data role.

Key Data Cleaning & Transformation Steps

The `Data cleaning Project.sql` script executes a multi-stage cleaning process, highlighting proficiency in complex SQL functions and data manipulation:

1. Standardize Date Format

- **Action:** Converted the `SaleDate` column from a datetime format to a consistent **standardized Date format**.
- **Techniques:** Used `CONVERT(Date, SaleDate)` and created a new column, `SaleDateConverted`, to store the clean date.

2. Populate Missing Data (NULLs)

- **Action:** Addressed missing (`NULL`) values in the `PropertyAddress` column.
- **Techniques:** Used a **Self-Join** on the table, matching records based on the unique `ParcelID` (since properties with the same `ParcelID` should have the same address) while ensuring the records were not the same row (`UniqueID` exclusion). The `ISNULL` function was used in an `UPDATE` statement to fill the missing addresses.

3. Break Out Addresses into Individual Columns

- **Action:** Separated concatenated address fields into distinct columns for easier analysis.
- **Property Address:** Split `PropertyAddress` into `PropertySplitAddress` (Street) and `PropertySplitCity` (City).
    - **Techniques:** Employed **string manipulation functions** like `SUBSTRING` and `CHARINDEX`.
- **Owner Address:** Split `OwnerAddress` (which includes address, city, and state) into `OwnerSplitAddress`, `OwnerSplitCity`, and `OwnerSplitState`.

    - **Techniques:** Used the powerful **PARSENAME** function along with **REPLACE** to efficiently parse the comma-separated data.

4. Standardize Categorical Fields

- **Action:** Ensured consistency in the SoldAsVacant column.
- **Techniques:** Used a **CASE statement** within an UPDATE to change 'Y' and 'N' values to the more explicit 'Yes' and 'No'.

5. Identify and Remove Duplicate Records

- **Action:** Identified and removed true duplicate rows that could skew analysis (based on key identifiers like ParcelID, PropertyAddress, SalePrice, SaleDate, and LegalReference).
- **Techniques:** Utilized a **Common Table Expression (CTE)** and the **ROW_NUMBER() OVER (PARTITION BY...) window function**—the industry-standard method for flagging duplicates.

6. Final Column Removal

- **Action:** Dropped the original, raw columns (OwnerAddress, TaxDistrict, PropertyAddress, and the original SaleDate) that were no longer needed after the successful creation of their cleaned/split counterparts, resulting in the final, clean dataset.