**Project 3 SQL Data Cleaning**

The following 12 SQL queries were used for data cleaning an Excel document uploaded to Microsoft Access. The queries were organized as follows:

* Query 1 tests that all data was successfully uploaded.
* Query 2 standardizes the date format.
* Queries 3-6 correct missing PropertyAddress values for multiple rows.
  + Through a quick test (query 3), you can see that each Parcel ID value corresponds to an address. Therefore, the missing property addresses can be found by finding the corresponding Parcel IDs and matching them to a missing property address.
  + Query 3 tests that the values are null.
  + Query 4 creates a new table new\_table.
  + Query 5 inserts values from the raw data to the new\_table.
  + Query 6 updates the original table with the necessary PropertyAddress values.
* Queries 7-9 change Y/N values to Yes/No to help with comprehension.
* Queries 10 and 11 remove duplicates.
  + Query 10 shows duplicates.
  + Between queries, I made a second version (v2) of my table by copy and pasting the “structure only” option. In the new table, I changed the Property Address, Sale Date, Sale Price, and Legal Reference to be primary keys. These are the indicators of a unique row.
  + Query 11 appended the old table’s data to the new data. The primary key change from the last step prevented duplicates from appending over because they were deemed key violations.
  + Only unique rows are in the second version (v2) of the table.
* Query 12 deletes unused columns.

**QUERY 1 (test data uploaded)**

SELECT \*

FROM NashvilleDataCleaning;

**QUERY 2 (standardize dates)**

UPDATE NashvilleDataCleaning SET SaleDate = CDate(SaleDate)

WHERE SaleDate;

**QUERY 3 (test PropertyAddress has null values)**

SELECT \*

FROM NashvilleDataCleaning

WHERE PropertyAddress is null

ORDER BY ParcelID;

**QUERY 4 (create new table)**

CREATE TABLE new\_table

(ParcelID varchar(50),

PropertyAddress varchar(255) NOT NULL

)

**QUERY 5 (insert values into table)**

INSERT INTO new\_table ( ParcelID, PropertyAddress )

SELECT NashvilleDataCleaning.ParcelID, NashvilleDataCleaning.PropertyAddress

FROM NashvilleDataCleaning

WHERE (((NashvilleDataCleaning.PropertyAddress) Is Not Null));

**QUERY 6 (update original table with new values)**

UPDATE NashvilleDataCleaning INNER JOIN new\_table ON NashvilleDataCleaning.ParcelID = new\_table.ParcelID SET new\_table.PropertyAddress = [NashvilleDataCleaning].[PropertyAddress];

**QUERY 7 (test some SoldAsVacant values are Y/N and some are Yes/No)**

SELECT DISTINCT (SoldAsVacant), COUNT(SoldAsVacant)

FROM NashvilleDataCleaning

GROUP BY SoldAsVacant

ORDER BY 2;

**QUERY 8 (change Y values to Yes values)**

UPDATE NashvilleDataCleaning SET SoldAsVacant = 'Yes'

WHERE SoldAsVacant = 'Y';

**QUERY 9 (change N values to No values)**

UPDATE NashvilleDataCleaning SET SoldAsVacant = 'No'

WHERE SoldAsVacant = 'N';

**QUERY 10 (show duplicates)**

SELECT NashvilleDataCleaning.PropertyAddress, NashvilleDataCleaning.SaleDate, NashvilleDataCleaning.SalePrice, NashvilleDataCleaning.LegalReference, NashvilleDataCleaning.UniqueID, NashvilleDataCleaning.ParcelID, NashvilleDataCleaning.LandUse, NashvilleDataCleaning.SoldAsVacant, NashvilleDataCleaning.OwnerName, NashvilleDataCleaning.OwnerAddress, NashvilleDataCleaning.Acreage, NashvilleDataCleaning.LandValue, NashvilleDataCleaning.BuildingValue, NashvilleDataCleaning.TotalValue, NashvilleDataCleaning.YearBuilt, NashvilleDataCleaning.Bedrooms, NashvilleDataCleaning.FullBath, NashvilleDataCleaning.HalfBath

FROM NashvilleDataCleaning

WHERE (((NashvilleDataCleaning.PropertyAddress) In (SELECT [PropertyAddress] FROM [NashvilleDataCleaning] As Tmp GROUP BY [PropertyAddress],[SaleDate],[SalePrice],[LegalReference] HAVING Count(\*)>1 And [SaleDate] = [NashvilleDataCleaning].[SaleDate] And [SalePrice] = [NashvilleDataCleaning].[SalePrice] And [LegalReference] = [NashvilleDataCleaning].[LegalReference])))

ORDER BY NashvilleDataCleaning.PropertyAddress, NashvilleDataCleaning.SaleDate, NashvilleDataCleaning.SalePrice, NashvilleDataCleaning.LegalReference;

**QUERY 11 (append non-duplicates to new table)**

INSERT INTO [NashvilleDataCleaning v2]

SELECT NashvilleDataCleaning.\*

FROM NashvilleDataCleaning;

**QUERY 12 (delete unused columns)**

ALTER TABLE NashvilleDataCleaning

DROP COLUMN TaxDistrict