**Nashville Housing Project SQL Code (PostgreSQL)**

\*/ Populate property address data \*/

SELECT \* FROM nashvillehousing

--WHERE propertyaddress IS NULL

ORDER BY ParcelID;

SELECT a.ParcelID, a.PropertyAddress, b.ParcelID,

b.PropertyAddress, COALESCE(a.PropertyAddress,b.PropertyAddress) FROM nashvillehousing a

JOIN nashvillehousing b ON

a.ParcelID = b.ParcelID

AND a.UniqueID != b.UniqueID

WHERE a.PropertyAddress IS NULL;

Update nashvillehousing

SET PropertyAddress = COALESCE(a.PropertyAddress,b.PropertyAddress)

FROM nashvillehousing a

JOIN nashvillehousing b ON

a.ParcelID = b.ParcelID

AND a.UniqueID != b.UniqueID

WHERE a.PropertyAddress IS NULL;

\*/ Splitting address into individual columns (address, city, state) \*/

SELECT

SUBSTRING(PropertyAddress,1,POSITION(',' in PropertyAddress)-1) AS Address,

SUBSTRING(PropertyAddress,POSITION(',' in PropertyAddress)+1,LENGTH(PropertyAddress))

AS Address

FROM nashvillehousing;

ALTER TABLE Nashvillehousing

ADD COLUMN PropertySplitAddress VARCHAR(255);

UPDATE Nashvillehousing

SET PropertySplitAddress = SUBSTRING(PropertyAddress,1,POSITION(',' in PropertyAddress)-1)

ALTER TABLE Nashvillehousing

ADD COLUMN PropertySplitCity VARCHAR(255);

UPDATE Nashvillehousing

SET PropertySplitCity = SUBSTRING(PropertyAddress,POSITION(',' in PropertyAddress)+1,LENGTH(PropertyAddress))

SELECT \* FROM nashvillehousing;

SELECT SPLIT\_PART(Owneraddress,',',1),

SPLIT\_PART(Owneraddress,',',2),

SPLIT\_PART(Owneraddress,',',3) FROM nashvillehousing;

ALTER TABLE nashvillehousing

ADD OwnerSplitAddress VARCHAR(255);

UPDATE nashvillehousing

SET OwnerSplitAddress = SPLIT\_PART(Owneraddress,',',1)

ALTER TABLE nashvillehousing

ADD OwnerSplitCity VARCHAR(255);

UPDATE nashvillehousing

SET OwnerSplitCity = SPLIT\_PART(Owneraddress,',',2)

ALTER TABLE nashvillehousing

ADD OwnerSplitState VARCHAR(255);

UPDATE nashvillehousing

SET OwnerSplitState = SPLIT\_PART(Owneraddress,',',3)

\*/ Change Y and N to Yes and No in “Sold as vacant” field \*/

SELECT soldasvacant, CASE

WHEN soldasvacant = 'Y' THEN 'Yes'

WHEN soldasvacant = 'N' THEN 'No'

ELSE soldasvacant

END FROM nashvillehousing;

UPDATE nashvillehousing

SET soldasvacant = CASE

WHEN soldasvacant = 'Y' THEN 'Yes'

WHEN soldasvacant = 'N' THEN 'No'

ELSE soldasvacant

END

\*/ Deleting Duplicates \*/

DELETE FROM nashvillehousing

WHERE UniqueID IN

(SELECT UniqueID FROM (

SELECT UniqueID, ROW\_NUMBER() OVER (

PARTITION BY ParcelID,

PropertyAddress,

SalePrice,

SaleDate,

LegalReference

ORDER BY UniqueID) row\_num

FROM nashvillehousing) t

WHERE t.row\_num >1);

\*/ Delete Unused Columns \*/

ALTER TABLE nashvillehousing

DROP COLUMN propertyaddress,

DROP COLUMN taxdistrict,

DROP COLUMN owneraddress;