# NashvilleHousing

# Executive Summary Report

## Data Set Anomalies

| **Key Value** | **Description of Anomaly** | **Plan for Resolution** |
| --- | --- | --- |
| 2045, all records affected  (UniqueID field) | Datatype is datetime  (SaleDate field) | Alter to Date datatype |
|  | Null values in PropertyAddress field | Populate Addresses from records with like ParcelID |
| 2045, all populated records affected (UniqueID field) | Compound data in PropertyAddress field | Parse into PropertyStreetAddress, and PropertyCity fields |
| 2045, all records affected  (UniqueID field) | Values set to ‘0’ for no and ‘1’ for yes in SoldAsVacant field | Update data to ‘N’ for No and ‘Y’ for yes |
| 28207, all populated records affected | Multiple name formats in OwnerName field | No action – Iterative correction necessary and will not increase functionality |
|  | Null values in OwnerAddress field | No action – Refer to research and assess data collection and source |
|  | Compound data in OwnerAddress field | Parse into OwnerStreetAddress, OwnerCity, and OwnerState |
| 29476, multiple records affected (UniqueID field) | NULL value in YearBuilt conficts with SoldAsVacant indicator | No Action - Refer to research and assess data collection and source |
| Multiple | Duplicate records found | Delete duplicate records |

## Data Types

| **Header Name From File** | **Data Types Note** |
| --- | --- |
| UniqueID | OK – int, not NULL |
| ParcelID | OK - nvarchar(50), not NULL |
| LandUse | OK - nvarchar(50), not NULL |
| SaleDate | DateTime, convert to Date |
| SalePrice | OK – money, not NULL |
| LegalReference | OK - nvarchar(50), not NULL |
| SoldAsVacant | OK – bit, not NULL |
| PropertyAddress | PARSE into PropertyStreetAddress (nvarchar(255)) and PropertyCity (nvarchar(50)) then DROP |
| OwnerName | OK – nvarchar(100) |
| OwnerAddress | OK – nvarchar(50) |
| Acreage | OK - float |
| TaxDistrict | OK - nvarchar(50) |
| LandValue | OK - int |
| BuildingValue | OK - int |
| TotalValue | OK - int |
| YearBuilt | OK - smallint |
| Bedrooms | OK - tinyint |
| FullBath | OK - tinyint |
| HalfBath | OK - tinyint |

## Specific Transformations Needed to Join the Data

There is not joining in the import as all data comes from a single file.

## Executive Summary

In review of the source data the overall approach will be the following:

1. Validate file data overall.
2. Review and clean each field as needed:
   1. Add fields
   2. Parse data
   3. Alter Data Types
   4. Correct anomalies as indicated in anomalies table
   5. Drop fields
3. Validate transformed data

Supporting SQL queries:

--Initial validation, 56477 records

SELECT COUNT(\*)

FROM dbo.NashvilleHousing

SELECT TOP 50 \*

FROM dbo.NashvilleHousing

-----UniqueID----------------------------------------------------------------------------------------------

-----Data Type bound to int, not NULL

--56477 distinct populated values, verified same as total record count

SELECT count(distinct(UniqueID))

FROM dbo.NashvilleHousing

-----ParcelID----------------------------------------------------------------------------------------------

-----Bound to nvarchar(50), not NULL

--48559 distinct populated values

SELECT count(distinct(ParcelID))

FROM dbo.NashvilleHousing

-----LandUse----------------------------------------------------------------------------------------------

--Bound to nvarchar(50), not NULL

--No cleaning necessary, queries below for data understanding

--Verify total distinct values

--39 distinct values

SELECT count(distinct(LandUse))

FROM dbo.NashvilleHousing

--Verify visually, no format or data issues apparent

SELECT distinct(LandUse)

FROM dbo.NashvilleHousing

-----SaleDate----------------------------------------------------------------------------------------------

--bound to NOT NULL

--Verify SaleDate DateTime vs Date

SELECT SaleDate, CONVERT(Date, SaleDate)

FROM dbo.NashvilleHousing

--Convert DateTime to Date

UPDATE NashvilleHousing

SET SaleDate = CONVERT(Date, SaleDate)

-----SalePrice---------------------------------------------------------------------------------------------

--Data Typle count ot Money, NOT NULL

--no cleaning necessary, queries below for data undertanding

--Verify distinct values

--8081 Distinct values exist

SELECT count(distinct(SalePrice))

FROM dbo.NashvilleHousing

--Data values do not indicate possible issues in range

SELECT min(SalePrice) as 'Minimum', MAX(SalePrice) as 'Maximum', AVG(SalePrice) as 'Average'

FROM dbo.NashvilleHousing

-----LegalReference---------------------------------------------------------------------------------------------

--Data Typle count ot Money, NOT NULL

--no cleaning necessary, queries below for data undertanding

--Verify distinct values

--52761 Distinct values exist

SELECT count(distinct(LegalReference))

FROM dbo.NashvilleHousing

--Data values do not indicate possible issues in formatting

SELECT LegalReference

FROM dbo.NashvilleHousing

--WHERE LegalReference not like '%-%'

-----SoldAsVacant------------------------------------------------------------------------------------------

--Verifiy data values and that no records are NULL

SELECT DISTINCT(SoldAsVacant)

FROM dbo.NashvilleHousing

--Verify new formatting correction format

--ELSE clause is not necessary in this case but is good form, nobody likes incomplete or 'crashy' code

SELECT SoldAsVacant,

CASE WHEN SoldAsVacant = '0' THEN 'N'

WHEN SoldAsVacant = '1' THEN 'Y'

ELSE SoldAsVacant

END

FROM dbo.NashvilleHousing

--Convert DataType to accept new data

ALTER TABLE NashvilleHousing

ALTER COLUMN SoldAsVacant char(1)

--Update data in table

UPDATE NashvilleHousing

SET SoldAsVacant =

CASE WHEN SoldAsVacant = '0' THEN 'N'

WHEN SoldAsVacant = '1' THEN 'Y'

ELSE SoldAsVacant

END

--Verify Update

SELECT DISTINCT(SoldAsVacant)

FROM NashvilleHousing

-----PropertyAddress--NULL Value correction-----------------------------------------------------------------

--Inspect field data

SELECT PropertyAddress

FROM dbo.NashvilleHousing

--Verify existence of NULL data in property address data

SELECT PropertyAddress

FROM dbo.NashvilleHousing

WHERE PropertyAddress is NULL

--Count NULL data instances in property address data

WITH PropAdd (PropertyAddress)

as

(

SELECT PropertyAddress

FROM dbo.NashvilleHousing

WHERE PropertyAddress is NULL

)

SELECT count(\*) as CountOfNull

FROM PropAdd

--Verified that duplicate ParcelID instances exist where both address values and null values exist

SELECT \*

FROM dbo.NashvilleHousing

WHERE PropertyAddress IS NULL

ORDER by ParcelID

--Expore NULL values with Address values on matching ParcelID values

SELECT a.ParcelID, a.PropertyAddress, b.ParcelID, b.PropertyAddress as PropAdd, ISNULL(a.PropertyAddress, b.PropertyAddress)

FROM dbo.NashvilleHousing a

JOIN dbo.NashvilleHousing b

on a.ParcelID = b.ParcelID

AND a.UniqueID <> b.UniqueID

WHERE a.PropertyAddress is NULL

--Update NULL values of PropertyAddress with correct address values

UPDATE a

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

FROM dbo.NashvilleHousing a

JOIN dbo.NashvilleHousing b

on a.ParcelID = b.ParcelID

AND a.UniqueID <> b.UniqueID

WHERE a.PropertyAddress is NULL

--Verified 0 records remaining with NULL in field.

SELECT COUNT(\*)

FROM dbo.NashvilleHousing

WHERE PropertyAddress IS NULL

------PropertyAddress----Formatting------------------------------------------------------------------------

--Verify address formatting as StreetAddress then comma then city

SELECT PropertyAddress

FROM dbo.NashvilleHousing

--Verify no space before comma in any records

SELECT PropertyAddress

FROM dbo.NashvilleHousing

WHERE PropertyAddress like '% ,%'

--Verify no space after comma in any records

SELECT PropertyAddress

FROM dbo.NashvilleHousing

WHERE PropertyAddress like '%, %'

--Verify new formatting correction format

SELECT

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

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

FROM dbo.NashvilleHousing

--Seperate Street Address and City into seperate fields

ALTER TABLE NashvilleHousing

ADD PropertyStreetAddress nvarchar(255)

ALTER TABLE NashvilleHousing

ADD PropertyCity nvarchar(255)

UPDATE NashvilleHousing

SET PropertyStreetAddress = SUBSTRING(PropertyAddress, 1, CHARINDEX(',',PropertyAddress)-1)

UPDATE dbo.NashvilleHousing

SET PropertyCity = SUBSTRING(PropertyAddress, CHARINDEX(',', PropertyAddress)+1, LEN(PropertyAddress))

--Verify Change

SELECT \*

FROM dbo.NashvilleHousing

--DROP PropertyAddress field no longer needed

ALTER TABLE NashvilleHousing

DROP COLUMN PropertyAddress

--Verify Change

SELECT \*

FROM dbo.NashvilleHousing

-----OwnerName-------------------------------------------------------------------------------------------

--No cleaning action taken per the below results

--100% of records have a UniqueID, SaleDate, SalePrice

--so all records have what can be considered primary use data

SELECT \*

FROM dbo.NashvilleHousing

WHERE UniqueID is NULL OR SaleDate is NULL or SalePrice is NULL

--Verify amount of records with NULL value as it appears large

--55.271% of records have NULL value in OwnerName

--Dropping records is not recommended due to primary data is available in records and total dropped records > 5%.

SELECT

AVG(CASE WHEN OwnerName is NULL THEN 1.0 ELSE 0 END)\*100 as '%NullCount',

AVG(CASE WHEN OwnerName is not NULL THEN 1.0 ELSE 0 END)\*100 as '%NotNullCount'

FROM dbo.NashvilleHousing

--94.234% of records with NULL values in OwnerName have a ParcelID with an alpha character inserted in the position after the 3rd number grouping

--this is a significant, but not absolute relationship

SELECT

AVG(CASE WHEN SUBSTRING(ParcelID,9,1) in ('A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z') THEN 1.0 ELSE 0 END)\*100 as '%AlphaCount',

AVG(CASE WHEN SUBSTRING(ParcelID,9,1) = '' THEN 1.0 ELSE 0 END)\*100 as '%NonAlphaCount'

FROM dbo.NashvilleHousing

WHERE OwnerName is NULL

--No records with non-NULL values in OwnerName have a ParcelID with an alpha character inserted in the position after the 3rd number grouping.

SELECT \*

FROM dbo.NashvilleHousing

--WHERE ParcelID = '018 12 0A 004.00' or ParcelID = '018 12 0 004.00'

WHERE OwnerName is not NULL and SUBSTRING(ParcelID,9,1) <> ''

ORDER BY ParcelID

--OwnerName contains multiple name formats including proper names with comma's, Company Names,

--and multiple names with and without middle initials, '&' 'jr' 'sr' and 'II' used, etc

SELECT \*

FROM dbo.NashvilleHousing

WHERE OwnerName not like '%,%'

-----OwnerAddress------------------------------------------------------------------------------------------

--Verify existance of NULL values

--30404 NULL records found

--There is no viable method to populate the data, record values being left NULL

SELECT count(\*)

FROM dbo.NashvilleHousing

WHERE OwnerAddress is NULL

--Verify address formatting as StreetAddress then comma then city

SELECT OwnerAddress

FROM dbo.NashvilleHousing

--Verify if space before comma in any records

SELECT OwnerAddress

FROM dbo.NashvilleHousing

WHERE OwnerAddress like '% ,%'

--Verify if space after comma in any records

SELECT OwnerAddress

FROM dbo.NashvilleHousing

WHERE OwnerAddress like '%, %'

--Verify record count of found Owner Address = '% ,%' of equals total record count

--Verify no space after comma in any records

SELECT count(OwnerAddress) as AddressWithSpaces

FROM dbo.NashvilleHousing

WHERE OwnerAddress like '%, %'

SELECT count(OwnerAddress) as TotalAddresses

FROM dbo.NashvilleHousing

--Verify new formatting correction format

SELECT

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

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

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

FROM dbo.NashvilleHousing

--Seperate Street Address, City, and State into seperate fields

ALTER TABLE NashvilleHousing

ADD OwnerStreetAddress nvarchar(255)

ALTER TABLE NashvilleHousing

ADD OwnerCity nvarchar(255)

ALTER TABLE NashvilleHousing

ADD OwnerState nvarchar(255)

UPDATE NashvilleHousing

SET OwnerStreetAddress = PARSENAME(REPLACE(OwnerAddress, ',','.'),3)

UPDATE dbo.NashvilleHousing

SET OwnerCity = PARSENAME(REPLACE(OwnerAddress, ',','.'),2)

UPDATE dbo.NashvilleHousing

SET OwnerState = PARSENAME(REPLACE(OwnerAddress, ',','.'),1)

--Verify Change

SELECT \*

FROM dbo.NashvilleHousing

--DROP PropertyAddress field no longer needed

ALTER TABLE NashvilleHousing

DROP COLUMN OwnerAddress

-----Acreage-----------------------------------------------------------------------------------------------

--Data Type bound to float, NULL is allowed

--no cleaning necessary, queries below for data undertanding

--Acreage has 519 distinct populated values, all float

SELECT count(distinct(Acreage))

FROM dbo.NashvilleHousing

--NULL values exist

SELECT Acreage

FROM dbo.NashvilleHousing

WHERE Acreage is NULL

--Data values do not indicate possible issues in format

SELECT min(Acreage) as 'Minimum', MAX(Acreage) as 'Maximum', AVG(Acreage) as 'Average'

FROM dbo.NashvilleHousing

--46.063% of values in Acreage are not null and the remaining 53.934% are NULL

--Same percentage values in Acreage, TaxDistrict, LandValue, BuildingValue, and TotalValue fields

SELECT

AVG(CASE WHEN Acreage is NOT NULL THEN 1.0 ELSE 0 END)\*100 as '%NotNULL',

AVG(CASE WHEN Acreage is NULL THEN 1.0 ELSE 0 END)\*100 as '%NULL'

FROM dbo.NashvilleHousing

-----TaxDistrict-----------------------------------------------------------------------------------------------

--Data Type bound to nvarchar(50), NULL is allowed

--no cleaning necessary, queries below for data undertanding

--TaxDistrict has 7 distinct populated values

SELECT count(distinct(TaxDistrict))

FROM dbo.NashvilleHousing

--Data values do not indicate possible issues in format

SELECT distinct(TaxDistrict)

FROM dbo.NashvilleHousing

--NULL values exist

SELECT TaxDistrict

FROM dbo.NashvilleHousing

WHERE TaxDistrict is NULL

--46.063% of values are not null and the remaining 53.934% are NULL

--Same percentage values in Acreage, TaxDistrict, LandValue, BuildingValue, and TotalValue fields

SELECT

AVG(CASE WHEN TaxDistrict is NOT NULL THEN 1.0 ELSE 0 END)\*100 as '%NotNULL',

AVG(CASE WHEN TaxDistrict is NULL THEN 1.0 ELSE 0 END)\*100 as '%NULL'

FROM dbo.NashvilleHousing

-----LandValue-----------------------------------------------------------------------------------------------

--Data Type bound to int, NULL is allowed

--no cleaning necessary, queries below for data undertanding

--LandValue has 1122 distinct populated values

SELECT count(distinct(LandValue))

FROM dbo.NashvilleHousing

--Data values do not indicate possible issues in format

SELECT min(LandValue) as 'Minimum', MAX(LandValue) as 'Maximum', AVG(LandValue) as 'Average'

FROM dbo.NashvilleHousing

--NULL values exist

SELECT LandValue

FROM dbo.NashvilleHousing

WHERE LandValue is NULL

--46.063% of values are not null and the remaining 53.934% are NULL

--Same percentage values in Acreage, TaxDistrict, LandValue, BuildingValue, and TotalValue fields

SELECT

AVG(CASE WHEN LandValue is NOT NULL THEN 1.0 ELSE 0 END)\*100 as '%NotNULL',

AVG(CASE WHEN LandValue is NULL THEN 1.0 ELSE 0 END)\*100 as '%NULL'

FROM dbo.NashvilleHousing

-----BuildingValue-----------------------------------------------------------------------------------------------

--Data Type bound to int, NULL is allowed

--no cleaning necessary, queries below for data undertanding

--BuildingValue has 4405 distinct populated values

SELECT count(distinct(BuildingValue))

FROM dbo.NashvilleHousing

--Data values do not indicate possible issues in format

SELECT min(BuildingValue) as 'Minimum', MAX(BuildingValue) as 'Maximum', AVG(CONVERT(decimal, BuildingValue)) as 'Average'

FROM dbo.NashvilleHousing

--NULL values exist

SELECT BuildingValue

FROM dbo.NashvilleHousing

WHERE BuildingValue is NULL

--46.063% of values are not null and the remaining 53.934% are NULL

--Same percentage values in Acreage, TaxDistrict, LandValue, BuildingValue, and TotalValue fields

SELECT

AVG(CASE WHEN BuildingValue is NOT NULL THEN 1.0 ELSE 0 END)\*100 as '%NotNULL',

AVG(CASE WHEN BuildingValue is NULL THEN 1.0 ELSE 0 END)\*100 as '%NULL'

FROM dbo.NashvilleHousing

-----TotalValue-----------------------------------------------------------------------------------------------

--Data Type bound to int, NULL is allowed

--no cleaning necessary, queries below for data undertanding

--TotalValue has 5848 distinct populated values

SELECT count(distinct(TotalValue))

FROM dbo.NashvilleHousing

--Data values do not indicate possible issues in format

SELECT min(TotalValue) as 'Minimum', MAX(TotalValue) as 'Maximum', AVG(CONVERT(decimal, TotalValue)) as 'Average'

FROM dbo.NashvilleHousing

--NULL values exist

SELECT TotalValue

FROM dbo.NashvilleHousing

WHERE TotalValue is NULL

--46.063% of values are not null and the remaining 53.934% are NULL

--Same percentage values in Acreage, TaxDistrict, LandValue, BuildingValue, and TotalValue fields

SELECT

AVG(CASE WHEN TotalValue is NOT NULL THEN 1.0 ELSE 0 END)\*100 as '%NotNULL',

AVG(CASE WHEN TotalValue is NULL THEN 1.0 ELSE 0 END)\*100 as '%NULL'

FROM dbo.NashvilleHousing

-----YearBuilt---------------------------------------------------------------------------------------------

--Data Type bound to int, NULL is allowed

--no cleaning necessary, queries below for data undertanding

--YearBuilt has 126 distinct populated values

SELECT count(distinct(YearBuilt))

FROM dbo.NashvilleHousing

--Data values do not indicate possible issues

SELECT min(YearBuilt) as 'Minimum', MAX(YearBuilt) as 'Maximum', AVG(CONVERT(decimal, YearBuilt)) as 'Average'

FROM dbo.NashvilleHousing

--NULL values exist

SELECT YearBuilt

FROM dbo.NashvilleHousing

WHERE YearBuilt is NULL

--42.784% of values are not null and the remaining 57.217% are NULL

SELECT

AVG(CASE WHEN YearBuilt is NOT NULL THEN 1.0 ELSE 0 END)\*100 as '%NotNULL',

AVG(CASE WHEN YearBuilt is NULL THEN 1.0 ELSE 0 END)\*100 as '%NULL'

FROM dbo.NashvilleHousing

--Percentage of Null values in YearBuilt does not match the percentage of properties that were not vacant

--91.718% when sold were not vacant but only 42.784% had a build date

--There is no programatic solution, recommending further research

SELECT

AVG(CASE WHEN SoldAsVacant = 'N' THEN 1.0 ELSE 0 END)\*100 as '%NotVacant',

AVG(CASE WHEN SoldAsVacant = 'Y' THEN 1.0 ELSE 0 END)\*100 as '%Vacant'

FROM dbo.NashvilleHousing

-----Bedrooms---------------------------------------------------------------------------------------------

--Data Type bound to int, NULL is allowed

--no cleaning necessary, queries below for data undertanding

--Bedrooms has 12 distinct populated values

SELECT count(distinct(Bedrooms))

FROM dbo.NashvilleHousing

--Data values do not indicate possible issues

SELECT min(Bedrooms) as 'Minimum', MAX(Bedrooms) as 'Maximum', AVG(CONVERT(decimal, Bedrooms)) as 'Average'

FROM dbo.NashvilleHousing

--NULL values exist

SELECT Bedrooms

FROM dbo.NashvilleHousing

WHERE Bedrooms is NULL

--42.773% of values are not null and the remainder are NULL

SELECT

AVG(CASE WHEN Bedrooms is NOT NULL THEN 1.0 ELSE 0 END)\*100 as '%NotNULL'

FROM dbo.NashvilleHousing

-----FullBath---------------------------------------------------------------------------------------------

--Data Type bound to int, NULL is allowed

--no cleaning necessary, queries below for data undertanding

--FullBath has 11 distinct populated values

SELECT count(distinct(FullBath))

FROM dbo.NashvilleHousing

--Data values do not indicate possible issues

SELECT min(FullBath) as 'Minimum', MAX(FullBath) as 'Maximum', AVG(CONVERT(decimal, FullBath)) as 'Average'

FROM dbo.NashvilleHousing

--NULL values exist

SELECT FullBath

FROM dbo.NashvilleHousing

WHERE FullBath is NULL

--42.982% of values are not null and the remainder are NULL

SELECT

AVG(CASE WHEN FullBath is NOT NULL THEN 1.0 ELSE 0 END)\*100 as '%NotNULL'

FROM dbo.NashvilleHousing

-----HalfBath---------------------------------------------------------------------------------------------

--Data Type bound to int, NULL is allowed

--no cleaning necessary, queries below for data undertanding

--HalfBath has 12 distinct populated values

SELECT count(distinct(HalfBath))

FROM dbo.NashvilleHousing

--Data values do not indicate possible issues

SELECT min(HalfBath) as 'Minimum', MAX(HalfBath) as 'Maximum', AVG(CONVERT(decimal, HalfBath)) as 'Average'

FROM dbo.NashvilleHousing

--NULL values exist

SELECT HalfBath

FROM dbo.NashvilleHousing

WHERE HalfBath is NULL

--42.773% of values are not null and the remainder are NULL

SELECT

AVG(CASE WHEN HalfBath is NOT NULL THEN 1.0 ELSE 0 END)\*100 as '%NotNULL'

FROM dbo.NashvilleHousing

-----RemoveDuplicates--------------------------------------------------------------------------------------

--Identify Duplicate records, 104 identified

WITH RowNumCTE

AS(

SELECT \*,

ROW\_NUMBER() OVER (

PARTITION BY ParcelID, PropertyStreetAddress, SalePrice, SaleDate, LegalReference

ORDER BY UniqueID ) row\_num

FROM dbo.NashvilleHousing

--ORDER BY ParcelID

)

--SELECT \*

SELECT COUNT(\*)

--DELETE

FROM RowNumCTE

WHERE row\_num > 1