```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [UniqueID ]
      ,[ParcelID]
      ,[LandUse]
      ,[PropertyAddress]
      ,[SaleDate]
      ,[SalePrice]
      ,[LegalReference]
      ,[SoldAsVacant]
      ,[OwnerName]
      ,[OwnerAddress]
      ,[Acreage]
      ,[TaxDistrict]
      ,[LandValue]
      ,[BuildingValue]
      ,[TotalValue]
      ,[YearBuilt]
      ,[Bedrooms]
      ,[FullBath]
      ,[HalfBath]
  FROM [Portfolio project].[dbo].[NashvilleHousing]



  /*

Cleaning Data in SQL Queries

*/


Select * From [Portfolio project]..[NashvilleHousing]


--------------------------------------------------------------------------------
  --------------------------------------

-- Standardize Date Format


Select saleDate, CONVERT(Date,SaleDate)
From [Portfolio project]..[NashvilleHousing]


Update [NashvilleHousing]
SET SaleDate = CONVERT(Date,SaleDate)

-- If it doesn't Update properly

ALTER TABLE [NashvilleHousing]
```

```sql
Add SaleDateConverted Date;

Update NashvilleHousing
SET SaleDateConverted = CONVERT(Date,SaleDate)




--------------------------------------------------------------------------------
--------------------------------------

-- Populate Property Address data

Select *
From [Portfolio project]..[NashvilleHousing]
--Where PropertyAddress is null
order by ParcelID



Select a.ParcelID, a.PropertyAddress, b.ParcelID, b.PropertyAddress, ISNULL
  (a.PropertyAddress,b.PropertyAddress)
From [Portfolio project]..[NashvilleHousing] a
JOIN [Portfolio project]..[NashvilleHousing] b
    on a.ParcelID = b.ParcelID
    AND a.[UniqueID ] <> b.[UniqueID ]
Where a.PropertyAddress is null


Update a
SET PropertyAddress = ISNULL(a.PropertyAddress,b.PropertyAddress)
From [Portfolio project]..[NashvilleHousing] a
JOIN [Portfolio project]..[NashvilleHousing] b
    on a.ParcelID = b.ParcelID
    AND a.[UniqueID ] <> b.[UniqueID ]
Where a.PropertyAddress is null




--------------------------------------------------------------------------------
  --------------------------------------

-- Breaking out Address into Individual Columns (Address, City, State)


Select PropertyAddress
From [Portfolio project]..[NashvilleHousing]
--Where PropertyAddress is null
--order by ParcelID
```

```sql
SELECT
SUBSTRING(PropertyAddress, 1, CHARINDEX(',', PropertyAddress) -1 ) as Address
, SUBSTRING(PropertyAddress, CHARINDEX(',', PropertyAddress) + 1 , LEN
  (PropertyAddress)) as City
From [Portfolio project]..[NashvilleHousing]


ALTER TABLE NashvilleHousing
Add PropertySplitAddress Nvarchar(255);

Update NashvilleHousing
SET PropertySplitAddress = SUBSTRING(PropertyAddress, 1, CHARINDEX(',',
  PropertyAddress) -1 )


ALTER TABLE NashvilleHousing
Add PropertySplitCity Nvarchar(255);

Update NashvilleHousing
SET PropertySplitCity = SUBSTRING(PropertyAddress, CHARINDEX(',', PropertyAddress)
  + 1 , LEN(PropertyAddress))




Select * From [Portfolio project]..[NashvilleHousing]




Select OwnerAddress
From [Portfolio project]..[NashvilleHousing]

Select
PARSENAME(REPLACE(OwnerAddress, ',', '.') , 3) address
,PARSENAME(REPLACE(OwnerAddress, ',', '.') , 2) city
,PARSENAME(REPLACE(OwnerAddress, ',', '.') , 1) state
From [Portfolio project]..[NashvilleHousing]


ALTER TABLE NashvilleHousing
Add OwnerSplitAddress Nvarchar(255);

Update NashvilleHousing
SET OwnerSplitAddress = PARSENAME(REPLACE(OwnerAddress, ',', '.') , 3)


ALTER TABLE NashvilleHousing
```

```sql
Add OwnerSplitCity Nvarchar(255);

Update NashvilleHousing
SET OwnerSplitCity = PARSENAME(REPLACE(OwnerAddress, ',', '.') , 2)



ALTER TABLE NashvilleHousing
Add OwnerSplitState Nvarchar(255);

Update NashvilleHousing
SET OwnerSplitState = PARSENAME(REPLACE(OwnerAddress, ',', '.') , 1)



Select *From [Portfolio project]..[NashvilleHousing]



--------------------------------------------------------------------------------
  -------------------------------------


-- Change Y and N to Yes and No in "Sold as Vacant" field


Select Distinct(SoldAsVacant), Count(SoldAsVacant) total
From [Portfolio project]..[NashvilleHousing]
Group by SoldAsVacant
order by 2




Select SoldAsVacant
, CASE When SoldAsVacant = 'Y' THEN 'Yes'
       When SoldAsVacant = 'N' THEN 'No'
       ELSE SoldAsVacant
       END
From [Portfolio project]..[NashvilleHousing]

Update NashvilleHousing
SET SoldAsVacant = CASE When SoldAsVacant = 'Y' THEN 'Yes'
       When SoldAsVacant = 'N' THEN 'No'
       ELSE SoldAsVacant
       END
```

```sql
--------------------------------------------------------------------------------  ⤶
  ----------------------------------------------------------------------

-- Remove Duplicates

WITH RowNumCTE AS(
Select *,
    ROW_NUMBER() OVER (
    PARTITION BY ParcelID,
                 PropertyAddress,
                 SalePrice,
                 SaleDate,
                 LegalReference
                 ORDER BY
                     UniqueID
                     ) row_num

From [Portfolio project]..[NashvilleHousing]
--order by ParcelID
)
Select *
From RowNumCTE
Where row_num > 1
Order by PropertyAddress




Select * From [Portfolio project]..[NashvilleHousing]




--------------------------------------------------------------------------------  ⤶
  ---------------------

-- Delete Unused Columns


Select * From [Portfolio project]..[NashvilleHousing]

ALTER TABLE [Portfolio project]..[NashvilleHousing]
DROP COLUMN OwnerAddress, TaxDistrict, PropertyAddress, SaleDate
```