

```
/* Cleaning Data with SQL */
```

```
SELECT *  
FROM `housing-data-project-418110.Nashville_housing_data.Nashville`  
LIMIT 1000;
```

```
/* Fix Date Format */
```

```
Select SaleDate, Cast(SaleDate as Date)  
FROM `housing-data-project-418110.Nashville_housing_data.Nashville` ;
```

```
-- Updating the table with the correct SaleDate format --  
Update `housing-data-project-418110.Nashville_housing_data.Nashville`  
Set SaleDate = Cast(SaleDate as Date);
```

```
/* Populate the Property Address Where N/A */
```

```
-- Looking to see if there are any N/A in PropertyAddress --  
Select PropertyAddress  
FROM `housing-data-project-418110.Nashville_housing_data.Nashville`  
Where PropertyAddress = 'N/A';
```

```
Select *  
FROM `housing-data-project-418110.Nashville_housing_data.Nashville`  
Order by ParcelID;
```

```
-- Self Joining table to figure out what N/A need to be populated to --  
Select a.ParcelID, a.PropertyAddress, b.ParcelID, b.PropertyAddress  
FROM `housing-data-project-418110.Nashville_housing_data.Nashville` a  
Join `housing-data-project-418110.Nashville_housing_data.Nashville` b  
ON a.ParcelID = b.ParcelID And a.UniqueID_ <> b.UniqueID_  
Where a.PropertyAddress = 'N/A';
```

```
-- Populated the Property Address that have N/A  
Select a.ParcelID, a.PropertyAddress, b.ParcelID, b.PropertyAddress,  
Case When a.PropertyAddress = 'N/A' Then b.PropertyAddress ELSE a.PropertyAddress END  
CorrectPropertyAddress  
FROM `housing-data-project-418110.Nashville_housing_data.Nashville` a  
Join `housing-data-project-418110.Nashville_housing_data.Nashville` b  
ON a.ParcelID = b.ParcelID And a.UniqueID_ <> b.UniqueID_  
Where a.PropertyAddress = 'N/A';
```

```
-- Updating the table with the populated Property Address  
Update a  
Set PropertyAddress= (Case When a.PropertyAddress = 'N/A' Then b.PropertyAddress ELSE  
a.PropertyAddress END )  
FROM `housing-data-project-418110.Nashville_housing_data.Nashville` a  
Join `housing-data-project-418110.Nashville_housing_data.Nashville` b
```

```

ON a.ParcelID = b.ParcelID And a.UniqueID_ <> b.UniqueID_
Where a.PropertyAddress = 'N/A';

-- Checking to see if CorrectPropertyAddress have N/A --
Select PropertyAddress
FROM `housing-data-project-418110.Nashville_housing_data.Nashville`
Where PropertyAddress = 'N/A';

/* Break Address into individual columns (Address, City, State)*/

-- Splitting Property Address into Address and City --
SELECT PropertyAddress,
Split(PropertyAddress, ',')[safe_offset(0)] as Address1,
Split(PropertyAddress, ',')[safe_offset(1)] as Address2
FROM `housing-data-project-418110.Nashville_housing_data.Nashville`;

-- Alter Table to add 2 columns for both newly split Property Address and City --
Alter TABLE `Nashville_housing_data.Nashville`
  Add Column Property_Address String,
  Add Column Property_City String ;

-- Update the table and populate the new column 'Address' --
Update `Nashville_housing_data.Nashville`
Set Property_Address = Split(PropertyAddress, ',')[safe_offset(0)];

-- Update the table and populate the new column 'City' --
Update `Nashville_housing_data.Nashville`
Set Property_City = Split(PropertyAddress, ',')[safe_offset(1)];

/* Separating Owner Address into Address, City and State */

SELECT OwnerAddress
FROM `housing-data-project-418110.Nashville_housing_data.Nashville`
LIMIT 1000;

-- Splitting Owner Address into Address, City, State --
SELECT OwnerAddress,
Split(OwnerAddress, ',')[safe_offset(0)] as Owner_Address,
Split(OwnerAddress, ',')[safe_offset(1)] as Owner_City,
RIGHT(OwnerAddress,2) as Owner_State
FROM `housing-data-project-418110.Nashville_housing_data.Nashville`;

-- Alter Table to add 3 columns for newly split Owner Address, City and State --
Alter TABLE `Nashville_housing_data.Nashville`
  Add Column Owner_Address String,
  Add Column Owner_City String,
  Add Column Owner_State String ;

```

```

-- Update the table and populate the new column 'Owner_Address' --
Update `Nashville_housing_data.Nashville`
Set Owner_Address = Split(OwnerAddress, ',')[safe_offset(0)];

-- Update the table and populate the new column 'Owner_City' --
Update `Nashville_housing_data.Nashville`
Set Owner_City = Split(OwnerAddress, ',')[safe_offset(1)];

-- Update the table and populate the new column 'Owner_State' --
Update `Nashville_housing_data.Nashville`
Set Owner_State = Right(OwnerAddress,2);

/* Change Sold As Vacancy column from 'true' and 'false' to 'yes' and 'no'*/

-- Check the unique values in the Sold as Vacant column --
Select Distinct(SoldAsVacant)
FROM `housing-data-project-418110.Nashville_housing_data.Nashville` ;

-- Change the Sold as Vacant column to a string data type --
Select SoldAsVacant, CAST(SoldAsVacant as string)
FROM `housing-data-project-418110.Nashville_housing_data.Nashville` ;

-- Updating the table with the correct SoldAsVacant format --
Update `housing-data-project-418110.Nashville_housing_data.Nashville`
Set SoldAsVacant = Cast(SoldAsVacant as string);

-- Changing the Sold As Vacant column to yes and no --
Select
Case when SoldAsVacant='true' then 'yes' when SoldAsVacant='false' then 'no' Else
SoldAsVacant End
FROM `housing-data-project-418110.Nashville_housing_data.Nashville` ;

-- Update the table to new Sold as Vacant values --
Update `Nashville_housing_data.Nashville`
Set SoldAsVacant = Case when SoldAsVacant='true' then 'yes' when SoldAsVacant='false'
then 'no' Else SoldAsVacant End;

```

```

/* Remove Duplicates*/

-- Create a CTE that will find duplicates --
With RowNumCTE As(
Select *,
    ROW_NUMBER() Over(
    PARTITION BY
        ParcelID,
        PropertyAddress,
        SalePrice,
        SaleDate,
        LegalReference
    ORDER BY
        UniqueID_) row_num
FROM `housing-data-project-418110.Nashville_housing_data.Nashville`
)

Select *
From RowNumCTE
Where row_num >1
ORDER BY PropertyAddress

-- With CTE, delete the duplicates --
With RowNumCTE As(
Select *,
    ROW_NUMBER() Over(
    PARTITION BY
        ParcelID,
        PropertyAddress,
        SalePrice,
        SaleDate,
        LegalReference
    ORDER BY
        UniqueID_) row_num
FROM `housing-data-project-418110.Nashville_housing_data.Nashville`
)

DELETE
From RowNumCTE
Where row_num >1;

/* Delete Unused Columns*/

ALTER TABLE `Nashville_housing_data.Nashville`
DROP COLUMN IF EXISTS OwnerAddress,
DROP COLUMN IF EXISTS PropertyAddress,
DROP COLUMN IF EXISTS TaxDistric;

```