Nashville Housing Data Cleaning Project

A series of SQL queries were executed to ensure the accuracy, consistency, and structural integrity of the "nashville_housing" dataset.

Key activities included:

1. Convert **sale_date** to date data type
2. Split **property_address** column into address and city columns
3. Split **owner_address** into address, city and state columns
4. In **sold_as_vacant**, change all ‘N’ values to ‘No’, change all ‘Y’ values to ‘Yes’
5. Remove duplicate rows

```sql
select *
from nashville_housing;
```

unique_id	parcel_id	land_use	property_address	sale_date	sale_price	legal_referance	sold_as_vacant	owner_name	owner_
2045	007 00 0 125.00	SINGLE FAMILY	1808 FOX CHASE DR, GOODLETTSVILLE	April 9, 2013	240000	20130412-0036474	No	FRAZIER, CYRENTHA LYNETTE	1808 F
16918	007 00 0 130.00	SINGLE FAMILY	1832 FOX CHASE DR, GOODLETTSVILLE	June 10, 2014	366000	20140619-0053768	No	BONER, CHARLES & LESLIE	1832 F
54582	007 00 0 138.00	SINGLE FAMILY	1864 FOX CHASE DR, GOODLETTSVILLE	September 26, 2016	435000	20160927-0101718	No	WILSON, JAMES E. & JOANNE	1864 F
43070	007 00 0 143.00	SINGLE FAMILY	1853 FOX CHASE DR, GOODLETTSVILLE	January 29, 2016	255000	20160129-0008913	No	BAKER, JAY K. & SUSAN E.	1853 F
22714	007 00 0 149.00	SINGLE FAMILY	1829 FOX CHASE DR, GOODLETTSVILLE	October 10, 2014	278000	20141015-0095255	No	POST, CHRISTOPHER M. & SAMANTHA C.	1829 F
18367	007 00 0 151.00	SINGLE FAMILY	1821 FOX CHASE DR, GOODLETTSVILLE	July 16, 2014	267000	20140718-0063802	No	FIELDS, KAREN L. & BRENT A.	1821 F
19804	007 14 0 002.00	SINGLE FAMILY	2005 SADIE LN, GOODLETTSVILLE	August 28, 2014	171000	20140903-0080214	No	HINTON, MICHAEL R. & CYNTHIA M. MOORE	2005 S
54583	007 14 0 024.00	SINGLE FAMILY	1917 GRACELAND DR, GOODLETTSVILLE	September 27, 2016	262000	20161005-0105441	No	BAILOR, DARRELL & TAMMY	1917 G
36500	007 14 0 026.00	SINGLE FAMILY	1428 SPRINGFIELD HWY, GOODLETTSVILLE	August 14, 2015	285000	20150819-0083440	No	ROBERTS, MISTY L. & ROBERT M.	1428 S
19805	007 14 0 034.00	SINGLE FAMILY	1420 SPRINGFIELD HWY, GOODLETTSVILLE	August 29, 2014	340000	20140909-0082348	No	LEE, JEFFREY & NANCY	1420 S
29467	007 14 0A 024...	SINGLE FAMILY	2209 KAYLA DR, GOODLETTSVILLE	April 14, 2015	425000	20150415-0033442	No	NULL	NULL
10754	007 14 0A 027...	SINGLE FAMILY	109 BAILEY VIEW CT, GOODLETTSVILLE	December 12, 2013	585000	20131227-0130352	No	NULL	NULL
34751	007 14 0B 010...	RESIDENTIAL...	1900 TINNIN RD, GOODLETTSVILLE	July 13, 2015	190000	20150717-0069947	No	NULL	NULL
4512	007 15 0 002.00	SINGLE FAMILY	629 GAYLEMORE DR, GOODLETTSVILLE	June 7, 2013	189900	20130612-0059715	No	URRUTIA, CARLOS MIGUEL & REBECCA	629 GA

nashville_housing 2 Read

Objective 1: Convert **sale_date** to date data type

```sql
ALTER TABLE nashville_housing
RENAME COLUMN sale_date TO old_sale_date;

ALTER TABLE nashville_housing
ADD COLUMN sale_date date;

UPDATE nashville_housing
SET sale_date = STR_TO_DATE(old_sale_date, '%M %d, %Y');

ALTER TABLE nashville_housing
DROP COLUMN old_sale_date;
```

New **sale_date**

sale_date
2013-04-09
2014-06-10
2016-09-26
2016-01-29
2014-10-10
2014-07-16
2014-08-28
2016-09-27
2015-08-14

Objective 2:

-- Split **property_address** column into address and city columns

unique_id	parcel_id	land_use	property_address	sale
2045	007 00 0 125.00	SINGLE FAMILY	1808 FOX CHASE DR, GOODLETTSVILLE	240
16918	007 00 0 130.00	SINGLE FAMILY	1832 FOX CHASE DR, GOODLETTSVILLE	366
54582	007 00 0 138.00	SINGLE FAMILY	1864 FOX CHASE DR, GOODLETTSVILLE	435
43070	007 00 0 143.00	SINGLE FAMILY	1853 FOX CHASE DR, GOODLETTSVILLE	255
22714	007 00 0 149.00	SINGLE FAMILY	1829 FOX CHASE DR, GOODLETTSVILLE	278
18367	007 00 0 151.00	SINGLE FAMILY	1821 FOX CHASE DR, GOODLETTSVILLE	267
19804	007 14 0 002.00	SINGLE FAMILY	2005 SADIE LN, GOODLETTSVILLE	171

```sql
SELECT
    SUBSTRING_INDEX(property_address, ',', 1) AS address,
    SUBSTRING_INDEX(property_address, ',', -1) AS city
FROM nashville_housing
```

Output:

address	city
1808 FOX CHASE DR	GOODLETTSVILLE
1832 FOX CHASE DR	GOODLETTSVILLE
1864 FOX CHASE DR	GOODLETTSVILLE
1853 FOX CHASE DR	GOODLETTSVILLE
1829 FOX CHASE DR	GOODLETTSVILLE
1821 FOX CHASE DR	GOODLETTSVILLE
2005 SADIE LN	GOODLETTSVILLE

-- in order to remove the initial space in city column, use trim function

```sql
SELECT
    SUBSTRING_INDEX(property_address, ',', 1) AS address,
    TRIM(LEADING ' ' FROM SUBSTRING_INDEX(property_address, ',', -1)) AS city
FROM nashville_housing
```

Output:

address	city
1808 FOX CHASE DR	GOODLETTSVILLE
1832 FOX CHASE DR	GOODLETTSVILLE
1864 FOX CHASE DR	GOODLETTSVILLE
1853 FOX CHASE DR	GOODLETTSVILLE
1829 FOX CHASE DR	GOODLETTSVILLE
1821 FOX CHASE DR	GOODLETTSVILLE
2005 SADIE LN	GOODLETTSVILLE

-- add new columns for the split address and city

```
91    ALTER TABLE nashville_housing
92    ADD COLUMN property_split_address VARCHAR(255);
93
94    ALTER TABLE nashville_housing
95    ADD COLUMN property_split_city VARCHAR(255);
```

-- Update **property_split_address** with the address from **property_address**

```
98    UPDATE nashville_housing
99    SET property_split_address = SUBSTRING_INDEX(property_address, ',', 1);
```

-- Update **property_split_city** with the city from **property_address**

```
105   UPDATE nashville_housing
106   SET property_split_city = TRIM(LEADING ' ' FROM SUBSTRING_INDEX(property_address, ',', -1));
```

Address formats after executing the above queries:

property_split_address	property_split_city
1808 FOX CHASE DR	GOODLETTSVILLE
1832 FOX CHASE DR	GOODLETTSVILLE
1864 FOX CHASE DR	GOODLETTSVILLE
1853 FOX CHASE DR	GOODLETTSVILLE
1829 FOX CHASE DR	GOODLETTSVILLE
1821 FOX CHASE DR	GOODLETTSVILLE
2005 SADIE LN	GOODLETTSVILLE

Objective 3:

Split **owner_address** into address, city and state columns

owner_address
1808 FOX CHASE DR, GOODLETTSVILLE, TN
1832 FOX CHASE DR, GOODLETTSVILLE, TN
1864 FOX CHASE DR, GOODLETTSVILLE, TN
1853 FOX CHASE DR, GOODLETTSVILLE, TN
1829 FOX CHASE DR, GOODLETTSVILLE, TN
1821 FOX CHASE DR, GOODLETTSVILLE, TN
2005 SADIE LN, GOODLETTSVILLE, TN

-- view column splits before updates

```
SELECT
    owner_address,
    SUBSTRING_INDEX(owner_address, ',', 1) AS owner_split_address,
    TRIM(LEADING ' ' FROM SUBSTRING_INDEX(SUBSTRING_INDEX(owner_address, ',', 2), ',', -1)) AS owner_split_city,
    TRIM(LEADING ' ' FROM SUBSTRING_INDEX(owner_address, ',' ,-1)) AS owner_split_state
FROM nashville_housing
```

Output:

owner_address	owner_split_address	owner_split_city	owner_split_st...
1808 FOX CHASE DR, GOODLETTSVILLE, TN	1808 FOX CHASE DR	GOODLETTSVILLE	TN
1832 FOX CHASE DR, GOODLETTSVILLE, TN	1832 FOX CHASE DR	GOODLETTSVILLE	TN
1864 FOX CHASE DR, GOODLETTSVILLE, TN	1864 FOX CHASE DR	GOODLETTSVILLE	TN
1853 FOX CHASE DR, GOODLETTSVILLE, TN	1853 FOX CHASE DR	GOODLETTSVILLE	TN
1829 FOX CHASE DR, GOODLETTSVILLE, TN	1829 FOX CHASE DR	GOODLETTSVILLE	TN
1821 FOX CHASE DR, GOODLETTSVILLE, TN	1821 FOX CHASE DR	GOODLETTSVILLE	TN
2005 SADIE LN, GOODLETTSVILLE, TN	2005 SADIE LN	GOODLETTSVILLE	TN

-- add columns for the split owner address, owner city, and owner state

```
ALTER TABLE nashville_housing
ADD owner_split_address VARCHAR(255),
ADD owner_split_city VARCHAR(255),
ADD owner_split_state VARCHAR(255);
```

-- update the new columns

```
UPDATE nashville_housing
SET owner_split_address = TRIM(SUBSTRING_INDEX(owner_address, ',', 1)),
    owner_split_city = TRIM(LEADING ' ' FROM SUBSTRING_INDEX(SUBSTRING_INDEX(owner_address, ',', 2), ',', -1)),
    owner_split_state = TRIM(SUBSTRING_INDEX(owner_address, ',', -1))
```

Address formats after executing the above queries:

owner_split_address	owner_split_city	owner_split_state
1808 FOX CHASE DR	GOODLETTSVILLE	TN
1832 FOX CHASE DR	GOODLETTSVILLE	TN
1864 FOX CHASE DR	GOODLETTSVILLE	TN
1853 FOX CHASE DR	GOODLETTSVILLE	TN
1829 FOX CHASE DR	GOODLETTSVILLE	TN
1821 FOX CHASE DR	GOODLETTSVILLE	TN
2005 SADIE LN	GOODLETTSVILLE	TN

Objective 4: in **sold_as_vacant**, change all 'N' values to 'No', change all 'Y' values to 'Yes'

```
-- identify the distinct values for sold_as_vacant, and count each distinct value occurrence
SELECT DISTINCT sold_as_vacant, COUNT(sold_as_vacant)
FROM nashville_housing
GROUP BY sold_as_vacant
```

Output:

sold_as_vacant	COUNT(sold_as_vacant)
N	51704
Y	4669

```
UPDATE nashville_housing
SET sold_as_vacant = (
    CASE
        WHEN lower(sold_as_vacant) = 'y' THEN 'Yes'
        WHEN lower(sold_as_vacant) = 'n' THEN 'No'
        ELSE sold_as_vacant
        END
);
```

Verify correct update:

```
SELECT DISTINCT sold_as_vacant, COUNT(sold_as_vacant)
FROM nashville_housing
GROUP BY sold_as_vacant;
```

Output:

sold_as_vacant	COUNT(sold_as_vacant)
No	51704
Yes	4669

Objective 5: remove duplicate rows

-- a duplicate row is defined by rows with same **parcel_id, property_address, sale_price, sale_date,** and **legal_reference**

-- first, identify if there are duplicate rows

```
SELECT
    unique_id, row_num
FROM
    (
        SELECT
            unique_id,
            ROW_NUMBER() OVER (PARTITION BY parcel_id, property_address, sale_price, sale_date, legal_reference ORDER BY unique_id) AS row_num
        FROM nashville_housing
    ) AS rank_rows
WHERE row_num > 1
```

Output:

unique_id	row_num
27116	2
27117	2
27118	2
27119	2
27120	2
27121	2
27122	2

Delete duplicate rows

```
DELETE
FROM nashville_housing
WHERE unique_id IN
    (
        SELECT
            unique_id
        FROM
            (
                SELECT
                    unique_id,
                    ROW_NUMBER() OVER (PARTITION BY parcel_id, property_address, sale_price, sale_date, legal_reference ORDER BY unique_id) AS row_num
                FROM nashville_housing
            ) AS rank_rows
        WHERE row_num > 1
    )
```

Verify no duplicates found:

```sql
SELECT unique_id
FROM
(
    SELECT
        unique_id,
        ROW_NUMBER() OVER (PARTITION BY parcel_id, property_address, sale_price, sale_date, legal_reference ORDER BY unique_id) AS row_num
    FROM nashville_housing
) AS rank_rows
WHERE row_num > 1
```

Output: