

# 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

```
1 select *
2 from nashville_housing;
```

unique_id	parcel_id	land_use	property_address	sale_date	sale_price	legal_reference	sold_as_vacant	owner_name	owner_id
2045	007 00 0 125.00	SINGLE FAMILY	1808 FOX CHASE DR, GOODLETTSVILLE	April 9, 2013	240000	20130412-0036474	No	FRAZIER, CYRENTA LYNETTE	1808 F
18918	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	20140716-0063802	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 C
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 024.00	SINGLE FAMILY	2209 KAYLA DR, GOODLETTSVILLE	April 14, 2015	425000	20150415-0033442	No		
10754	007 14 0A 027.00	SINGLE FAMILY	109 BAILEY VIEW CT, GOODLETTSVILLE	December 12, 2013	585000	20131227-0130352	No		
34751	007 14 0B 010.00	RESIDENTIAL...	1900 TINNIN RD, GOODLETTSVILLE	July 13, 2015	190000	20150717-0069947	No		
4512	007 15 0 002.00	SINGLE FAMILY	629 GAYLEMORE DR, GOODLETTSVILLE	June 7, 2013	189900	20130612-0058715	No	URRUTIA, CARLOS MIGUEL & REBECCA	629 G
10045	007 15 0 002.00	SINGLE FAMILY	1004 CAMERON DR, GOODLETTSVILLE	June 28, 2014	473000	20140628-0058650	No	CALEDONIA ALMA L. & EDUARDO A. R.	1004 C

## Objective 1: Convert `sale_date` to date data type

```
4 ALTER TABLE nashville_housing
5   RENAME COLUMN sale_date TO old_sale_date;
6
7 ALTER TABLE nashville_housing
8   ADD COLUMN sale_date date;
9
10 UPDATE nashville_housing
11   SET sale_date = STR_TO_DATE(old_sale_date, '%M %d, %Y');
12
13 ALTER TABLE nashville_housing
14   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
2014-08-28

## 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	2400
	16918	007 00 0 130.00	SINGLE FAMILY	1832 FOX CHASE DR, GOODLETTSVILLE	3660
	54582	007 00 0 138.00	SINGLE FAMILY	1864 FOX CHASE DR, GOODLETTSVILLE	4350
	43070	007 00 0 143.00	SINGLE FAMILY	1853 FOX CHASE DR, GOODLETTSVILLE	2550
	22714	007 00 0 149.00	SINGLE FAMILY	1829 FOX CHASE DR, GOODLETTSVILLE	2780
	18367	007 00 0 151.00	SINGLE FAMILY	1821 FOX CHASE DR, GOODLETTSVILLE	2670
	19804	007 14 0 002.00	SINGLE FAMILY	2005 SADIE LN. GOODLETTSVILLE	1710

```
85  SELECT
86      SUBSTRING_INDEX(property_address, ',', 1) AS address,
87      SUBSTRING_INDEX(property_address, ',', -1) AS city
88  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 leading space in city column, use trim function

```
83  SELECT
84      SUBSTRING_INDEX(property_address, ',', 1) AS address,
85      TRIM(LEADING ' ' FROM SUBSTRING_INDEX(property_address, ',', -1)) AS city
86  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);
96
```

-- 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

```
115 • SELECT
116     owner_address,
117     SUBSTRING_INDEX(owner_address, ',', 1) AS owner_split_address,
118     TRIM(LEADING ' ' FROM SUBSTRING_INDEX(SUBSTRING_INDEX(owner_address, ',', 2), ',', -1)) AS owner_split_city,
119     TRIM(LEADING ' ' FROM SUBSTRING_INDEX(owner_address, ',', -1)) AS owner_split_state
120 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

```
123 • ALTER TABLE nashville_housing
124     ADD owner_split_address VARCHAR(255),
125     ADD owner_split_city VARCHAR(255),
126     ADD owner_split_state VARCHAR(255);
```

-- update the new columns

```
129 • UPDATE nashville_housing
130     SET owner_split_address = TRIM(SUBSTRING_INDEX(owner_address, ',', 1)),
131         owner_split_city = TRIM(LEADING ' ' FROM SUBSTRING_INDEX(SUBSTRING_INDEX(owner_address, ',', 2), ',', -1)),
132         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

```
153 • SELECT DISTINCT sold_as_vacant, COUNT(sold_as_vacant)
154 FROM nashville_housing
155 GROUP BY sold_as_vacant
```

Output:

	sold_as_vacant	COUNT(sold_as_vacant)
	N	51704
	Y	4669

```
150 • UPDATE nashville_housing
151 ⊖ SET sold_as_vacant = (
152 ⊖ CASE
153     WHEN lower(sold_as_vacant) = 'y' THEN 'Yes'
154     WHEN lower(sold_as_vacant) = 'n' THEN 'No'
155     ELSE sold_as_vacant
156     END
157 ⊖ );
```

Verify correct update:

```
159 • SELECT DISTINCT sold_as_vacant, COUNT(sold_as_vacant)
160 FROM nashville_housing
161 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

```
167 SELECT
168     unique_id, row_num
169 FROM
170     (
171         SELECT
172             unique_id,
173             ROW_NUMBER() OVER (PARTITION BY parcel_id, property_address, sale_price, sale_date, legal_reference ORDER BY unique_id) AS row_num
174         FROM nashville_housing
175     ) AS rank_rows
176 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:

```
194 SELECT unique_id
195 FROM
196 (
197     SELECT
198         unique_id,
199         ROW_NUMBER() OVER (PARTITION BY parcel_id, property_address, sale_price, sale_date, legal_reference ORDER BY unique_id) AS row_num
200     FROM nashville_housing
201 ) AS rank_rows
202 WHERE row_num > 1
```

Output:

unique_id