

Document highlight:

- Tool Installation
- Database and Table creation, including dataset import
 - Nashville Housing dataset sampling
- Data Cleansing in MySQL (query and results):
 - Cleaning Data in using SQL queries
 - Standardize Date Format
 - Populate Property Address data
 - Breaking out Address into Individual Columns (Address, City, State)
 - Change Y and N to Yes and No in "Sold as Vacant" field
 - Remove Duplicates
 - Delete Unused Columns

1. Setup MySQL Workbench (community)

Download and install (follow installation guide):

<https://dev.mysql.com/downloads/installer/>

The screenshot shows the MySQL Community Downloads page for the MySQL Installer 8.0.40. The page has a dark header with the MySQL logo and the text "MySQL Community Downloads". Below the header, there is a breadcrumb trail: "MySQL Installer". The main content area is titled "MySQL Installer 8.0.40" and contains a note: "Note: MySQL 8.0 is the final series with MySQL Installer. As of MySQL 8.1, use a MySQL product's MSI or Zip archive for installation. MySQL Server 8.1 and higher also bundle MySQL Configurator, a tool that helps configure MySQL Server." Below the note, there are two dropdown menus: "Select Version:" with "8.0.40" selected, and "Select Operating System:" with "Microsoft Windows" selected. Below these, there is a table with two rows of download links. The first row is for "Windows (x86, 32-bit), MSI Installer" (8.0.40, 2.1 M) with a "Download" button. The second row is for "Windows (x86, 32-bit), MSI Installer" (8.0.40, 306.4 M) with a "Download" button. Below the table, there is a note: "We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download." At the bottom of the page, there is a footer with the Oracle logo and the text "© 2024 Oracle". Below the footer, there is a link to "Privacy / Do Not Sell My Info | Terms of Use | Trademark Policy | Cookie Preferences".

General Availability (GA) Releases Archives

MySQL Installer 8.0.40

Note: MySQL 8.0 is the final series with MySQL Installer. As of MySQL 8.1, use a MySQL product's MSI or Zip archive for installation. MySQL Server 8.1 and higher also bundle MySQL Configurator, a tool that helps configure MySQL Server.

Select Version:
8.0.40

Select Operating System:
Microsoft Windows

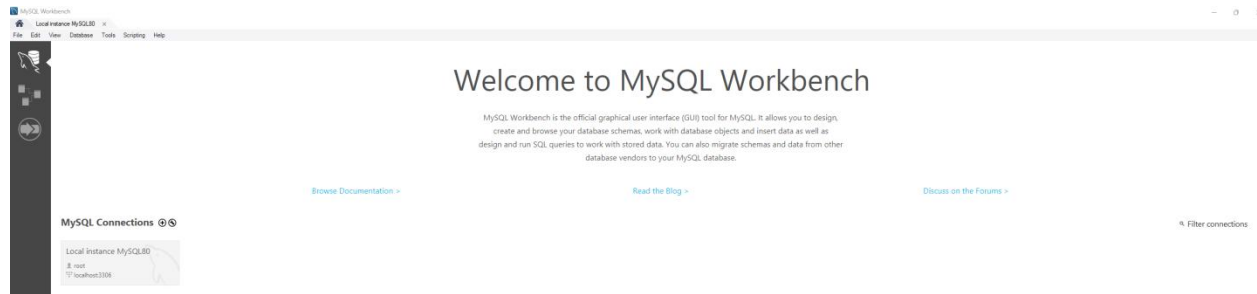
Windows (x86, 32-bit), MSI Installer (mysql-installer-web-community-8.0.40.0.msi)	8.0.40	2.1 M	Download
Windows (x86, 32-bit), MSI Installer (mysql-installer-community-8.0.40.0.msi)	8.0.40	306.4 M	Download

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

ORACLE © 2024 Oracle


[Privacy / Do Not Sell My Info](#) | [Terms of Use](#) | [Trademark Policy](#) | [Cookie Preferences](#)

MySQL home screen:



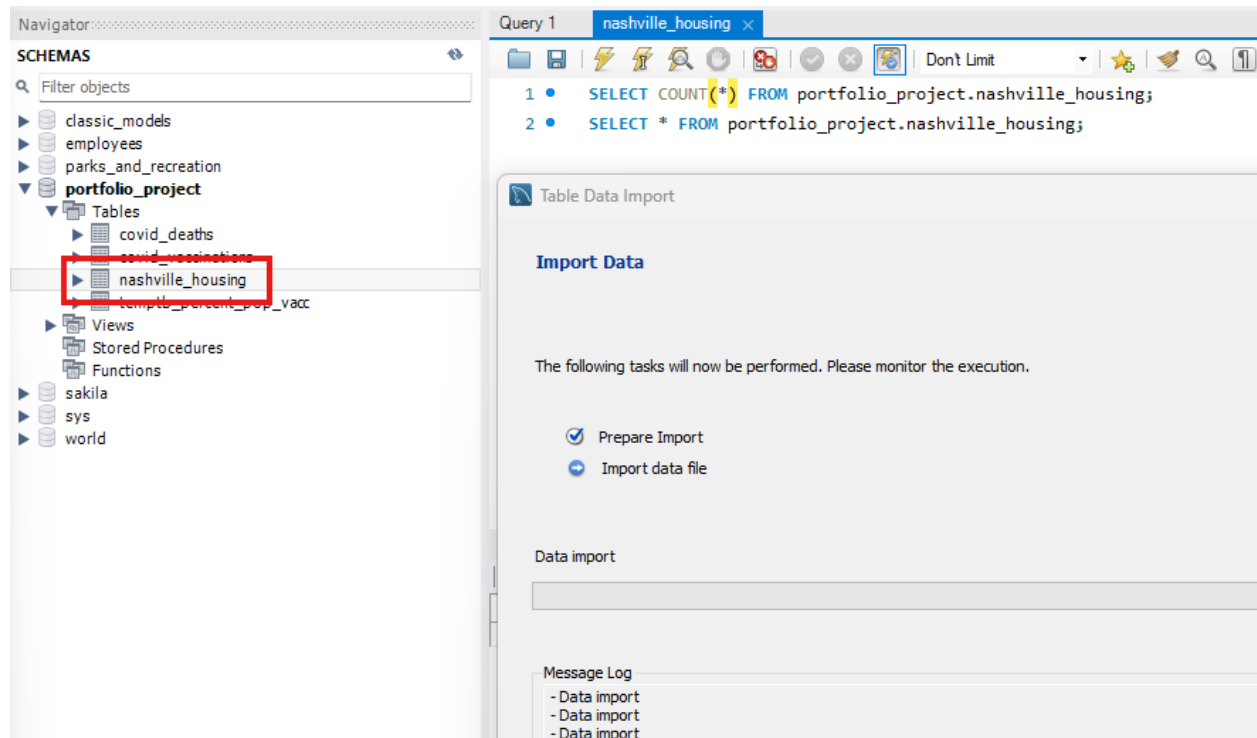
2. Create a database (create tables and load the set of data into it)

Sampling dataset:

 Nashville Housing Data for Data Cleaning CSV

2024-10-31 12:18 PM

Create a database and import the file (going into the table) in MySQL Workbench:



3. Quick query run to ensure that (1) tables were created and (2) data were imported

Navigator: SCHEMAS

- Filter objects
- classic_models
- employees
- parks_and_recreation
- portfolio_project
 - Tables
 - covid_deaths
 - covid_vaccinations
 - nashville_housing
 - temptb_percent_pop_vacc
 - Views
 - Stored Procedures
 - Functions

Query 1: nashville_housing

```
1 • SELECT COUNT(*) FROM portfolio_project.nashville_housing;
2 → SELECT * FROM portfolio_project.nashville_housing;
```

Result Grid

	UniqueID	ParcelID	LandUse	PropertyAddress	SaleDate	SalePrice	LegalReference	SoldAsVacant
▶	2045	007 00 0 125.00	SINGLE FAMILY	1808 FOX CHASE DR, GOODLETTSVILLE	April 9, 2013	240000	20130412-0036474	No
	16918	007 00 0 130.00	SINGLE FAMILY	1832 FOX CHASE DR, GOODLETTSVILLE	June 10, 2014	366000	20140619-0053768	No
	54582	007 00 0 138.00	SINGLE FAMILY	1864 FOX CHASE DR, GOODLETTSVILLE	September 26, 2016	435000	20160927-0101718	No

4. Cleaning Data in using SQL queries

02-Data Cleaning queries in My...

```
1 /*
2
3 Cleaning Data in SQL Queries
4
5 */
6
7 -- retrieve loaded data
8 • SELECT
9     *
10 FROM
11     portfolio_project.nashville_housing
12 ;
13
```

Result Grid

	UniqueID	ParcelID	LandUse	PropertyAddress	SaleDate	SalePrice	LegalReference	SoldAsVacant	OwnerName
▶	2045	007 00 0 125.00	SINGLE FAMILY	1808 FOX CHASE DR, GOODLETTSVILLE	2013-04-09 0:00	240000	20130412-0036474	No	FRAZIER, CYRE
	16918	007 00 0 130.00	SINGLE FAMILY	1832 FOX CHASE DR, GOODLETTSVILLE	2014-06-10 0:00	366000	20140619-0053768	No	BONER, CHARLI
	54582	007 00 0 138.00	SINGLE FAMILY	1864 FOX CHASE DR, GOODLETTSVILLE	2016-09-26 0:00	435000	20160927-0101718	No	WILSON, JAMES
	43070	007 00 0 143.00	SINGLE FAMILY	1853 FOX CHASE DR, GOODLETTSVILLE	2016-01-29 0:00	255000	20160129-0008913	No	BAKER, JAY K.
	22714	007 00 0 149.00	SINGLE FAMILY	1829 FOX CHASE DR, GOODLETTSVILLE	2014-10-10 0:00	278000	20141015-0095255	No	POST, CHRISTOPHER
	18367	007 00 0 151.00	SINGLE FAMILY	1821 FOX CHASE DR, GOODLETTSVILLE	2014-07-16 0:00	267000	20140718-0063802	No	FIELDS, KAREN

25 09:04:29 SELECT * FROM portfolio_project.nashville_housing 24007 row(s) returned 0.000 sec / 0.079 sec

5. Standardize Date Format

```
02-Data Cleaning queries in My...
15
16 -- Standardize Date Format
17
18 -- build and test the query...
19 • SELECT
20     n_haus.SaleDate AS 'Original Sale Date',
21     -- n_haus.SaleDate_conv,
22     CONVERT(n_haus.SaleDate, date) AS 'Converted Sale Date'
23 FROM
24     portfolio_project.nashville_housing as n_haus
25 ;
```

	Original Sale Date	Converted Sale Date
▶	2013-04-09 0:00	2013-04-09
	2014-06-10 0:00	2014-06-10
	2016-09-26 0:00	2016-09-26
	2016-01-29 0:00	2016-01-29
	2014-10-10 0:00	2014-10-10
	2014-07-16 0:00	2014-07-16

Add the table columns

```
27 -- create a column and dump the SaleDate values (converted into standard date format)...
28
29 -- alter the table and add column
30 • ALTER TABLE nashville_housing
31 ADD SaleDate_conv date;
--
```

After alter table

◆	HalfBath
◆	SaleDate_conv

Update the added table column and load the data

```
02-Data Cleaning queries in My... x
[Icons] Don't Limit
33  -- update the table column and load with converted data
34  • UPDATE nashville_housing
35    SET SaleDate_conv = convert(SaleDate, date);
36
37  -- retrieve loaded data
38  • SELECT
39    SaleDate_conv
40  FROM
41    nashville_housing
42  ;
43
```

After updating the added table column

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
SaleDate_conv				
2013-04-09				
2014-06-10				
2016-09-26				
2016-01-29				
2014-10-10				
2014-07-16				
2014-08-28				

6. Populate Property Address Data

```
02-Data Cleaning queries in My... x
[Icons] Don't Limit
46  -- Populate Property Address data if NULL
47
48  -- retrieve data...
49  • SELECT
50    n_haus.UniqueID,
51    n_haus.ParcelID,
52    n_haus.PropertyAddress
53  FROM
54    portfolio_project.nashville_housing as n_haus
55  WHERE
56    n_haus.PropertyAddress IS NULL
57  ORDER BY
58    n_haus.ParcelID
59  ;
```

Before update

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	UniqueID	ParcelID	PropertyAddress
	6993	015 14 0 060.00	NULL
	8903	018 00 0 164.00	NULL
	27163	018 07 0 109.00	NULL

02-Data Cleaning queries in My...

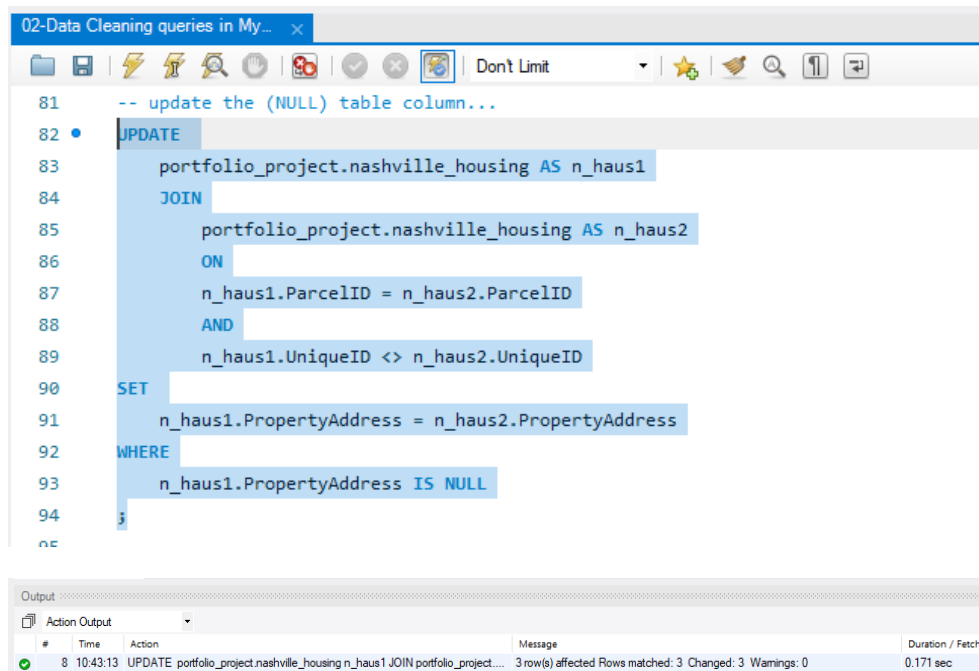
Don't Limit

```
61 -- build and test the query...
62 SELECT
63     n_haus1.ParcelID,
64     n_haus1.UniqueID,
65     n_haus1.PropertyAddress AS 'Destination Address',
66     n_haus2.ParcelID,
67     n_haus2.UniqueID,
68     n_haus2.PropertyAddress AS 'Source Address'
69 FROM
70     portfolio_project.nashville_housing n_haus1
71 JOIN
72     portfolio_project.nashville_housing n_haus2
73 ON
74     n_haus1.ParcelID = n_haus2.ParcelID
75 AND
76     n_haus1.UniqueID <> n_haus2.UniqueID
77 WHERE
78     n_haus1.PropertyAddress IS NULL
79
```

Columns to be updated (with Null values)

	ParcelID	UniqueID	Destination Address	ParcelID	UniqueID	Source Address
	015 14 0 060.00	6993	NULL	015 14 0 060.00	43071	3113 MILLIKEN DR, JOELTON
	018 00 0 164.00	8903	NULL	018 00 0 164.00	16921	332 MONCRIEF AVE, GOODLETTSVILLE
	018 07 0 109.00	27163	NULL	018 07 0 109.00	34756	105 BRYAN HOUSE DR, GOODLETTSVILLE

Update query



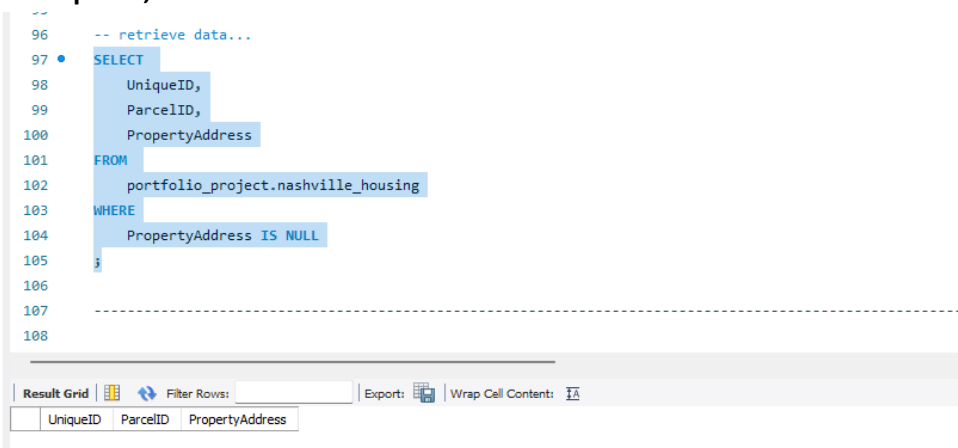
The screenshot shows a SQL IDE window titled "02-Data Cleaning queries in My...". The query editor contains an UPDATE statement. The output pane shows the execution results.

```
81 -- update the (NULL) table column...
82 UPDATE
83     portfolio_project.nashville_housing AS n_haus1
84 JOIN
85     portfolio_project.nashville_housing AS n_haus2
86 ON
87     n_haus1.ParcelID = n_haus2.ParcelID
88 AND
89     n_haus1.UniqueID <> n_haus2.UniqueID
90 SET
91     n_haus1.PropertyAddress = n_haus2.PropertyAddress
92 WHERE
93     n_haus1.PropertyAddress IS NULL
94 ;
```

Output:

#	Time	Action	Message	Duration / Fetch
8	10:43:13	UPDATE portfolio_project.nashville_housing n_haus1 JOIN portfolio_project...	3 row(s) affected Rows matched: 3 Changed: 3 Warnings: 0	0.171 sec

After update, no data with Null values



The screenshot shows a SQL IDE window with a SELECT query. The result grid below the query shows the data returned.

```
96 -- retrieve data...
97 SELECT
98     UniqueID,
99     ParcelID,
100     PropertyAddress
101 FROM
102     portfolio_project.nashville_housing
103 WHERE
104     PropertyAddress IS NULL
105 ;
```

Result Grid:

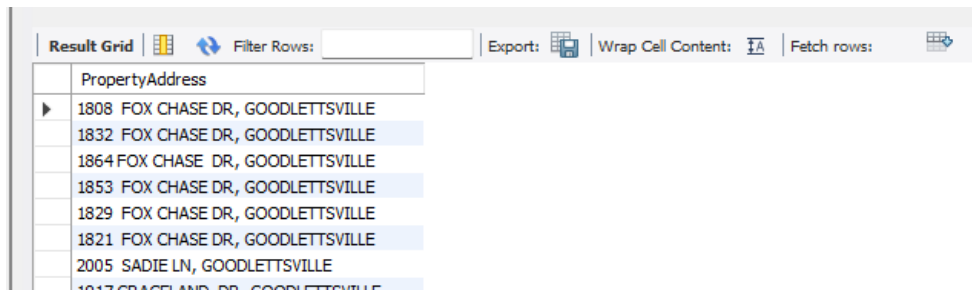
UniqueID	ParcelID	PropertyAddress
----------	----------	-----------------

7. Breaking out Address into Individual Columns (Address, City, State)

Split the PROPERTY address column value

```
109 -- Breaking out Address into Individual Columns (Address, City, State)
110
111 -- *****Split the PROPERTY address column value*****
112
113 -- retrieve data...
114 • SELECT
115     PropertyAddress
116 FROM
117     portfolio_project.nashville_housing
118 ;
119
```

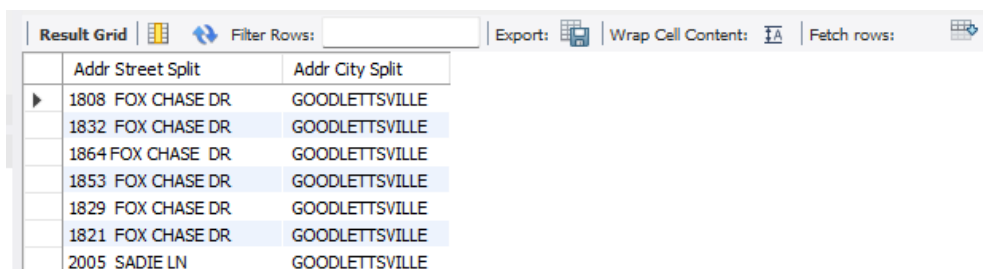
Column to be updated (split the address)



The screenshot shows a database interface with a toolbar at the top containing 'Result Grid', 'Filter Rows', 'Export', 'Wrap Cell Content', and 'Fetch rows'. Below the toolbar is a table with one column, 'PropertyAddress'. The table contains several rows of addresses, including '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', and '2005 SADIE LN, GOODLETTSVILLE'.

PropertyAddress
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

```
120 -- Split the property address and have it as (1) STREET and (2) CITY
121 -- build and test the query...
122 • SELECT
123     SUBSTRING(PropertyAddress, 1, LOCATE(',', PropertyAddress) -1) AS 'Addr Street Split',
124     SUBSTRING(PropertyAddress, LOCATE(',', PropertyAddress) +1) AS 'Addr City Split'
125 FROM
126     portfolio_project.nashville_housing
127 ;
128
```



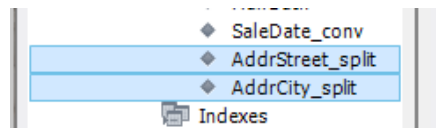
The screenshot shows the same database interface as before, but the result grid now has two columns: 'Addr Street Split' and 'Addr City Split'. The data is split from the previous 'PropertyAddress' column. For example, '1808 FOX CHASE DR' is in the first column and 'GOODLETTSVILLE' is in the second column.

Addr Street Split	Addr City Split
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 the table columns and load the data

```
130 • ALTER TABLE nashville_housing
131 ADD AddrStreet_split varchar(255);
132
133 -- update the table column and load with converted data
134 • UPDATE nashville_housing
135 SET AddrStreet_split = SUBSTRING(PropertyAddress, 1, LOCATE(',', PropertyAddress) -1)
136 ;
137
138 -- alter the table and add column
139 • ALTER TABLE nashville_housing
140 ADD AddrCity_split varchar(255);
141
142 -- update the table column and load with converted data
143 • UPDATE nashville_housing
144 SET AddrCity_split = SUBSTRING(PropertyAddress, LOCATE(',', PropertyAddress) +1)
145 ;
```

After alter table



After updating the added table columns

```
✓ 45 12:46:39 UPDATE nashville_housing SET AddrStreet_split = SUBSTRING(PropertyAddress, 1, LOCATE(',', PropertyAddress) -1) 24007 row(s) affected Rows matched: 24007 Changed: 24007 Warnings: 0
✓ 46 12:46:47 UPDATE nashville_housing SET AddrCity_split = SUBSTRING(PropertyAddress, LOCATE(',', PropertyAddress) +1) 24007 row(s) affected Rows matched: 24007 Changed: 24007 Warnings: 0
```

After column creation and load of split of data

```
144 -- retrieve loaded data
145 • SELECT
146     AddrStreet_split,
147     AddrCity_split
148 FROM
149     nashville_housing
150 ;
151
152 -----
153
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
AddrStreet_split	AddrCity_split			
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			
1917 GRACELAND DR	GOODLETTSVILLE			
1478 SPRINGFIELD BLVD	GOODLETTSVILLE			

Split the OWNER address column value

```

154
155      -- *****Split the OWNER address column value*****
156
157      -- retrieve data...
158 •    SELECT
159          OwnerAddress
160      FROM
161          portfolio_project.nashville_housing
162      ;
163

```

Column to be updated (split the address)

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
OwnerAddress				
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				

```

164      -- Split the address and have it as (1) Street and (2) City
165      -- build and test the query...
166 •    SELECT
167          SUBSTRING(OwnerAddress, 1, LOCATE(',', OwnerAddress) - 1) AS 'OAddr Street Split',
168          TRIM(SUBSTRING_INDEX(SUBSTRING_INDEX(OwnerAddress, ',', 2), ',', -1)) AS 'OAddr City Split',
169          TRIM(SUBSTRING_INDEX(OwnerAddress, ',', -1)) AS 'OAddr State Split'
170      FROM
171          portfolio_project.nashville_housing
172      ;
173

```

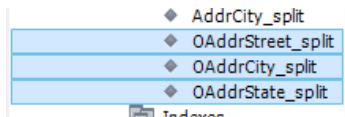
Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
OAddr Street Split	OAddr City Split	OAddr State Split		
1808 FOX CHASE DR	GOODLETTSVILLE	TN		
1832 FOX CHASE DR	GOODLETTSVILLE	TN		
1864 FOX CHASE DR	GOODLETTSVILLE	TN		
1853 FOX CHASE DR	GOODLETTSVILLE	TN		
1829 FOX CHASE DR	GOODLETTSVILLE	TN		
1821 FOX CHASE DR	GOODLETTSVILLE	TN		

Add the table columns and load the data

```
174 -- alter the table and add column (STREET)
175 • ALTER TABLE nashville_housing
176 ADD OAddrStreet_split varchar(255);
177
178 -- update the table column and load with converted data
179 • UPDATE nashville_housing
180 SET OAddrStreet_split = SUBSTRING(OwnerAddress, 1, LOCATE(',', OwnerAddress) -1)
181 ;
182
183 -- alter the table and add column (CITY)
184 • ALTER TABLE nashville_housing
185 ADD OAddrCity_split varchar(255);
186
187 -- update the table column and load with converted data
188 • UPDATE nashville_housing
189 SET OAddrCity_split = TRIM(SUBSTRING_INDEX(SUBSTRING_INDEX(OwnerAddress, ',', 2), ',', -1))
190 ;
191
192 -- alter the table and add column (STATE)
193 • ALTER TABLE nashville_housing
194 ADD OAddrState_split varchar(255);
195
196 -- update the table column and load with converted data
197 • UPDATE nashville_housing
198 SET OAddrState_split = TRIM(SUBSTRING_INDEX(OwnerAddress, ',', -1))
199 ;
```

After alter table

✓	67	14:13:11	ALTER TABLE nashville_housing ADD OAddrStreet_split varchar(255)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec
✓	68	14:13:16	ALTER TABLE nashville_housing ADD OAddrCity_split varchar(255)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.062 sec
✓	69	14:13:19	ALTER TABLE nashville_housing ADD OAddrState_split varchar(255)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec



After updating the added table columns

✓	70	14:14:21	UPDATE nashville_housing SET OAddrStreet_split = SUBSTRING(OwnerAddress, 1, LOCATE(',', OwnerAddress) -1)	24007 row(s) affected Rows matched: 24007 Changed: 24007 Warnings: 0	0.985 sec
✓	71	14:14:26	UPDATE nashville_housing SET OAddrCity_split = TRIM(SUBSTRING_INDEX(SUBSTRING_INDEX(OwnerAddress, ',', 2), ',', -1))	24007 row(s) affected Rows matched: 24007 Changed: 24007 Warnings: 0	0.938 sec
✓	72	14:14:31	UPDATE nashville_housing SET OAddrState_split = TRIM(SUBSTRING_INDEX(OwnerAddress, ',', -1))	24007 row(s) affected Rows matched: 24007 Changed: 24007 Warnings: 0	0.875 sec

After column creation and load of split of data

```
02-Data Cleaning queries in My... x
[Icons] | Don't Limit
201 -- retrieve loaded data
202 • SELECT
203     OAddrStreet_split,
204     OAddrCity_split,
205     OAddrState_split
206 FROM
207     nashville_housing
208 ;
209 -----
```

Result Grid			
Filter Rows: <input type="text"/>			
Export:			
Wrap Cell Content:			
Fetch rows:			
	OAddrStreet_split	OAddrCity_split	OAddrState_split
▶	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 SAGE LN	GOODLETTSVILLE	TN

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

```

02-Data Cleaning queries in My... x
[Icons] Don't Limit [Icons]
210
211 -- Change Y and N to Yes and No in "Sold as Vacant" field
212
213 -- retrieve data...
214 • SELECT
215     DISTINCT(SoldAsVacant),
216     COUNT(SoldAsVacant)
217 FROM
218     portfolio_project.nashville_housing
219 GROUP BY
220     SoldAsVacant
221 ORDER BY
222     2
223 ;

```

Result Grid		
Filter Rows: <input type="text"/>		
Export:		
Wrap Cell Content:		
	SoldAsVacant	COUNT(SoldAsVacant)
▶	Y	21
	N	151
	Yes	398
	No	23437

```

225 -- change any 'Y' to 'Yes' and 'N' to 'No'
226 • SELECT
227     SoldAsVacant,
228     CASE
229         WHEN SoldAsVacant = 'Y' THEN 'Yes'
230         WHEN SoldAsVacant = 'N' THEN 'No'
231         ELSE SoldAsVacant
232     END AS 'Updated Sold As Vacant'
233 FROM
234     nashville_housing
235 WHERE SoldAsVacant = 'Y'
236 ;
237

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	SoldAsVacant	Updated Sold As Vacant
▶	Y	Yes
	Y	Yes
	Y	Yes

```

225 -- change any 'Y' to 'Yes' and 'N' to 'No'
226 • SELECT
227     SoldAsVacant,
228     CASE
229         WHEN SoldAsVacant = 'Y' THEN 'Yes'
230         WHEN SoldAsVacant = 'N' THEN 'No'
231         ELSE SoldAsVacant
232     END AS 'Updated Sold As Vacant'
233 FROM
234     nashville_housing
235 WHERE SoldAsVacant = 'N'
236 ;
237

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	SoldAsVacant	Updated Sold As Vacant
▶	N	No
	N	No
	N	No

After updating the added table columns

```
238 -- update the table column and load with converted data
239 • UPDATE nashville_housing
240 SET SoldAsVacant =
241 CASE
242 WHEN SoldAsVacant = 'Y' THEN 'Yes'
243 WHEN SoldAsVacant = 'N' THEN 'No'
244 ELSE SoldAsVacant
245 END
246 ;
247
```

After updating the data

```
-- retrieve data...
• SELECT
    DISTINCT(SoldAsVacant),
    COUNT(SoldAsVacant)
FROM
    portfolio_project.nashville_housing
GROUP BY
    SoldAsVacant
ORDER BY
    2
;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
SoldAsVacant	COUNT(SoldAsVacant)			
Yes	419			
No	23588			

9. Remove Duplicates

Sampling of records with duplicates

```
-- Remove Duplicates

-- retrieve data (sampling of duplicates)...
• SELECT
    *
FROM
    portfolio_project.nashville_housing
WHERE
    ParcelID = '107 14 0 157.00'
;
```

Result Grid											
Filter Rows:											
Export: Wrap Cell Contents											
UniqueID	ParcelID	LandUse	PropertyAddress	SaleDate	SalePrice	LegalReference	SoldAsVacant	OwnerName	OwnerAddress	Acreage	
26505	107 14 0 157.00	SINGLE FAMILY	1003 BRILEY PKWY, NASHVILLE	2015-02-12 0:00	80000	20150219-0014347	No	GREEN, KASSANDRA	1003 BRILEY PKWY, NASHVILLE, TN	0.22	
27354	107 14 0 157.00	SINGLE FAMILY	1003 BRILEY PKWY, NASHVILLE	2015-02-12 0:00	80000	20150219-0014347	No	GREEN, KASSANDRA	1003 BRILEY PKWY, NASHVILLE, TN	0.22	

Display all records with duplicates

```

2  -- using CTE (DISPLAY all duplicates)...
3  WITH RowNumCTE AS (
4      SELECT
5          *,
6          ROW_NUMBER() OVER(
7              PARTITION BY
8                  ParcelID,
9                  PropertyAddress,
10                 SalePrice,
11                 SaleDate,
12                 LegalReference
13             ORDER BY
14                 UniqueID
15             ) AS RowNum
16  FROM
17      portfolio_project.nashville_housing
18  ORDER BY
19      ParcelID
20  )
21  -- with CTE (sort of a temp table), RETRIEVE data from it (duplicates)
22  SELECT
23      *
24  FROM
25      RowNumCTE
26  WHERE
27      RowNum > 1
28  ORDER BY
29      PropertyAddress
30  ;

```

Result Grid											
Filter Rows:											
Export: Wrap Cell Contents											
UniqueID	ParcelID	LandUse	PropertyAddress	SaleDate	SalePrice	LegalReference	SoldAsVacant	OwnerName	OwnerAddress	Acreage	
27354	107 14 0 157.00	SINGLE FAMILY	1003 BRILEY PKWY, NASHVILLE	2015-02-12 0:00	80000	20150219-0014347	No	GREEN, KASSANDRA	1003 BRILEY PKWY		
27383	150 04 0 144.00	SINGLE FAMILY	1005 ESTURBRIDGE CT, ANTIOCH	2015-02-13 0:00	180000	20150213-0013947	No	TINNEY, STU	1005 ESTURBRIDGE		
27362	119 16 0 084.00	DUPLEX	1006 DRUMMOND DR, NASHVILLE	2015-02-17 0:00	80000	20150218-0014086	No	POST, KEVIN	1006 DRUMMOND DR		
27132	091 08 0 056.00	SINGLE FAMILY	1011 43RD AVE N, NASHVILLE	2015-02-23 0:00	69900	20150227-0017102	No	MENDING HEARTS, INC.	1011 43RD AVE N,		
27123	091 04 0 046.00	SINGLE FAMILY	1038 40TH AVE N, NASHVILLE	2015-02-23 0:00	40000	20150227-0016978	No	DAVIS, CRYSTAL ANN	1038 40TH AVE N,		
27363	120 05 0 183.00	SINGLE FAMILY	1304 WINTHORNE DR, NASHVILLE	2015-02-12 0:00	130000	20150218-0013626	No	CASTILLO, KEVIN E. & RIVERA, MARA	1304 WINTHORNE		
27116	081 15 0 263.00	SINGLE FAMILY	1520 14TH AVE N, NASHVILLE	2015-02-12 0:00	55000	20150218-0013742	No	SUMMEY, CLARENCE	1520 14TH AVE N,		

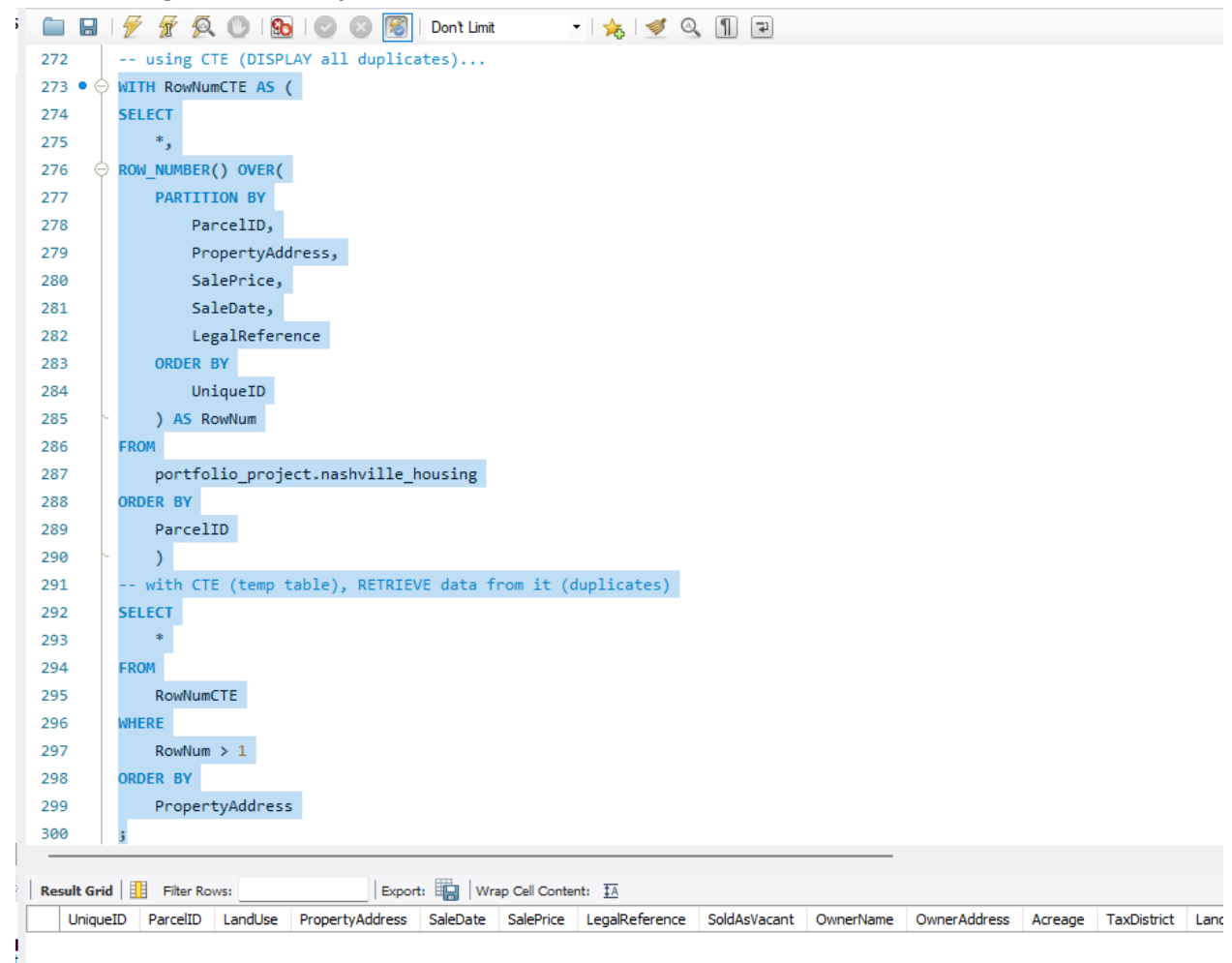
Delete all duplicate records

```
-- using CTE (DELETE all duplicates)...
WITH RowNumCTE AS
(
  SELECT
    *,
    ROW_NUMBER() OVER(
      PARTITION BY
        ParcelID,
        PropertyAddress,
        SalePrice,
        SaleDate,
        LegalReference
      ORDER BY
        UniqueID
    ) AS RowNum
  FROM
    portfolio_project.nashville_housing
)
-- with CTE, JOIN with the main table to be able to DELETE the duplicate data
DELETE
FROM
  nashville_housing
USING
  nashville_housing
JOIN
  RowNumCTE
ON
  nashville_housing.ParcelID = RowNumCTE.ParcelID
WHERE
  RowNum > 1
;
```

102 15:39:17 WITH RowNumCTE AS (SELECT *, ROW_NUMBER() OVER(PARTITION BY ParcelID, Pr... 97 row(s) affected

2.734 sec

After deleting, check the duplicates



```
272 -- using CTE (DISPLAY all duplicates)...
273 WITH RowNumCTE AS (
274     SELECT
275         *,
276         ROW_NUMBER() OVER(
277             PARTITION BY
278                 ParcelID,
279                 PropertyAddress,
280                 SalePrice,
281                 SaleDate,
282                 LegalReference
283             ORDER BY
284                 UniqueID
285         ) AS RowNum
286     FROM
287         portfolio_project.nashville_housing
288     ORDER BY
289         ParcelID
290 )
291 -- with CTE (temp table), RETRIEVE data from it (duplicates)
292 SELECT
293     *
294 FROM
295     RowNumCTE
296 WHERE
297     RowNum > 1
298 ORDER BY
299     PropertyAddress
300 ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

UniqueID	ParcelID	LandUse	PropertyAddress	SaleDate	SalePrice	LegalReference	SoldAsVacant	OwnerName	OwnerAddress	Acreage	TaxDistrict	Lanc
----------	----------	---------	-----------------	----------	-----------	----------------	--------------	-----------	--------------	---------	-------------	------

10. Delete Unused Columns

```
-- Delete Unused Columns (part of the data cleanup for this exercise)

-- retrieve data...
SELECT
    PropertyAddress,
    OwnerAddress,
    SaleDate,
    TaxDistrict
FROM
    portfolio_project.nashville_housing
;
```

Result Grid					Filter Rows:	Export:	Wrap Cell Contents:	Fetch rows:
	PropertyAddress	OwnerAddress	SaleDate	TaxDistrict				
▶	1808 FOX CHASE DR, GOODLETTSVILLE	1808 FOX CHASE DR, GOODLETTSVILLE, TN	2013-04-09 0:00	GENERAL SERVICES DISTRICT				
	1832 FOX CHASE DR, GOODLETTSVILLE	1832 FOX CHASE DR, GOODLETTSVILLE, TN	2014-06-10 0:00	GENERAL SERVICES DISTRICT				
	1864 FOX CHASE DR, GOODLETTSVILLE	1864 FOX CHASE DR, GOODLETTSVILLE, TN	2016-09-26 0:00	GENERAL SERVICES DISTRICT				
	1853 FOX CHASE DR, GOODLETTSVILLE	1853 FOX CHASE DR, GOODLETTSVILLE, TN	2016-01-29 0:00	GENERAL SERVICES DISTRICT				
	1829 FOX CHASE DR, GOODLETTSVILLE	1829 FOX CHASE DR, GOODLETTSVILLE, TN	2014-10-10 0:00	GENERAL SERVICES DISTRICT				
	1821 FOX CHASE DR, GOODLETTSVILLE	1821 FOX CHASE DR, GOODLETTSVILLE, TN	2014-07-16 0:00	GENERAL SERVICES DISTRICT				
	2005 SADIE LN, GOODLETTSVILLE	2005 SADIE LN, GOODLETTSVILLE, TN	2014-08-28 0:00	GENERAL SERVICES DISTRICT				
	1917 GRACELAND DR, GOODLETTSVILLE	1917 GRACELAND DR, GOODLETTSVILLE, TN	2016-09-27 0:00	GENERAL SERVICES DISTRICT				

Drop (unused) column tables

```
-- delete columns
• ALTER TABLE portfolio_project.nashville_housing
  DROP COLUMN PropertyAddress,
  DROP COLUMN OwnerAddress,
  DROP COLUMN SaleDate,
  DROP COLUMN TaxDistrict;
```

✓ 116 16:16:08 ALTER TABLE portfolio_project.nashville_housing DROP COLUMN PropertyAd... 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.047 sec

After dropping the table columns

```
-- retrieve data...
• SELECT
  PropertyAddress,
  OwnerAddress,
  SaleDate,
  TaxDistrict
FROM portfolio_project.nashville_housing;
```

✗ 117 16:17:13 SELECT PropertyAddress, OwnerAddress, SaleDate, TaxDistrict FROM p... Error Code: 1054. Unknown column 'PropertyAddress' in 'field list' 0.016 sec

*****END*****