Data Cleaning with PopSQL/MySQL Workbench

We’ll be using MySQL for Data cleaning as it is currently more practical for this specific purpose than BigQuery

First, we must prepare our dataset by filling in blank spaces with NULL (otherwise Workbench will have trouble processing all rows):

Graphical user interface, application

Description automatically generated with medium confidence

To

Table

Description automatically generated

By

Cmd + A (to select all rows), Ctrl + A (windows)

Next

Shift + Cmd + H (Mac)

Ctrl + H (Windows)

A screen shot of a computer

Description automatically generated with medium confidence

Click Replace All

A picture containing text, monitor, cellphone, electronics

Description automatically generated

Now save file as CSV format from Excel, and you’re good to go

Table

Description automatically generated

Uploading data to database via Workbench

First, you’ll have to create a database if you haven’t already or if there are none and use it (I will personally drop my database and table for this demonstration):

A screenshot of a computer

Description automatically generated with medium confidence

By querying DROP DATABASE housing; (What I’ll be using for this Project)

Run it:

A screenshot of a computer

Description automatically generated with medium confidence

Now create new database with CREATE DATABASE housing; (or pick any name)

Run it

A screenshot of a computer

Description automatically generated with medium confidence

Now right click Refresh and click on Tables:

Graphical user interface, text, application, chat or text message

Description automatically generated

Select correct/updated file by selecting Browse…

Graphical user interface, text, application

Description automatically generated

Like this:

A screenshot of a computer

Description automatically generated

Then click Next

Graphical user interface, text, application

Description automatically generated

Configure table, I created New and named it nashvillehousing, Click Next

A screenshot of a computer

Description automatically generated

Configure Datatypes for MySQL here, then click Next

A screenshot of a computer screen

Description automatically generated with medium confidence

Ready to go, click Next

Graphical user interface, text

Description automatically generated

Monitor Process (takes a few minutes depending on size of dataset)

Graphical user interface, application

Description automatically generated

Once done importing, click Next

Graphical user interface, application

Description automatically generated

Done, now we can start Analyzing data

Graphical user interface, text

Description automatically generated

I’ll be using PopSQL for this project (ready to go)

Graphical user interface

Description automatically generated

Let’s run dates

SELECT SaleDate

FROM nashvillehousing

LIMIT 100000;

Graphical user interface, text, application, email

Description automatically generated

Must do my best to translate from MSSQL Server to MySQL functions (instructor uses Microsoft SQL server not yet supported for Mac M1 ARM architecture, may be able to use using container with Docker once updated ARM image becomes available)

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

Select PropertyAddress

From nashvillehousing;

-- had to find charvar equivalent, LOCATE or INSTR

SELECT

SUBSTRING(PropertyAddress, 1, LOCATE(',', PropertyAddress) -1 ) as Address

, SUBSTRING(PropertyAddress, LOCATE(',', PropertyAddress) + 1 , CHAR\_LENGTH(PropertyAddress)) as Address

From NashvilleHousing;

RUN IT

Graphical user interface, text

Description automatically generated

Now we’ll write the following queries to add updated info. Into our database (mysql)

ALTER TABLE nashvilleHousing

Add NewPropertySplitAddress NVARCHAR(255);

Update nashvillehousing

SET NewPropertySplitAddress = SUBSTRING(PropertyAddress, 1, LOCATE(',', PropertyAddress) -1 )

--

ALTER TABLE nashvillehousing

Add NewPropertySplitCity NVARCHAR(255);

Update nashvillehousing

SET NewPropertySplitCity = SUBSTRING(PropertyAddress, LOCATE(',', PropertyAddress) + 1 , CHAR\_LENGTH(PropertyAddress))

SELECT \* FROM nashvillehousing limit 100000;

Run it

And now we can see new columns we created [27:51]

Table

Description automatically generated

Eliminate duplicate rows

Table

Description automatically generated

Now,

Graphical user interface

Description automatically generated