# **Nashville Housing Project**

This project is a SQL and Python project to analyse trends in the sale price data for houses in Nashville. The idea for the project was originated by Alex the Analyst on his YouTube channel and further expanded upon to build a predictive model in python. There are two parts to the project, the first part is a data cleaning of the data in SQL of which not a lot of wrangling was done as more of the cleaning and manipulation will be done in Python. The second part of the project is to build a regression model to analyse which variables contribute most to the value of the house

This project is for personal development and to gain more experience working on end-to-end projects. It is not a research project, thus the format in the coding and in this document will not reflect any research formatting.

**THE DATA:** The data used in this project is provided by Alex the Analyst from his Github pages. The data is in a csv file format. This data was loaded onto SQL SSMS using the object Explorer onto my project database

# Part 1: Data Wrangling on SQL

The first part I want to do is have an overall view of the data to see what I’m working with and what kind of cleaning, manipulation and/or transformation. This can also be done using the objective explorer which will give us a breakdown of the data type and how much rows of data we have.

There are 56477 rows of data with some columns having nulls which I will chose to deal with on an as-and-when case when dealing with each column of interest to me.

There first bit of missing values is the property address information. What I noticed is that some of the properties were being sold multiple times meaning that the UniqueID and ParcelID entries, which are unique for each property was repeated, thus we can create a view from which we extract the address to update our table using a SELF JOIN (SELF JOIN was used to create a table in which the available address for the matching UniqueID and ParcelID can be viewed to extract)

Once this data was obtained, I separated the street address, city and state information can be separated in case I want to use those as dummy variables in my regression model.

Our objective is to build regression models using our dataset thus the next bit of manipulation is to look for and remove duplicate entries in our dataset as this will create a problem by giving extra weight to that particular data point (this can be extra harmful if this data point is an outlier or influencer in our statistical analysis). Thus I created a CTE (common table expression) where I partitioned the rows containing duplicate information in the ParcelID, PropertyAddress, SalePrice, SaleDate, LegalReference columns to rank them so that I can delete all entries with a row rank higher than 1 by querying this CTE.

There seems to be null values still for the OwnerName and OwnerAddress, acreage, land value & other quantitative data that will confuse my regression model thus let’s explore them & see how we deal with them.

The first I will do is order by the fields with null values and see if information can't be pulled from a previous transaction that might allow us to update the missing values. In exploring the data further, I noticed that most of the missing values originates from missing values in the YearBuilt column thus I filtered the columns where the YearBuilt column had null values, nd this was 30404 rows of data (over 60% of our data). This is a huge portion of the data thus a suitable solution is to replace the missing values with updated information. This could possibly be obtained from Nashville’s housing department however for the sake of simplicity and building a regression model I chose to delete them (in Python) as we still have over 20000 rows of data to work with

I also noticed a trend in the average sale price per year when grouping the data by the YearBuilt thus a useful column to have would be the Age of the House. I also deleted the string columns in which would not be useful to the model such as TaxDistrict.

**Note:** Some of the techniques used were not optimal however I just want to showcase my ability to query useful information and display my ability to update and manipulate datasets thus I did not change the methods used. There are notes available on the SQL project file explaining as well

The final table was exported as Nashville House cleaned, both csv files will be loaded onto my Github page.

# Part 2: Building Regression Model in Python

The regression analysis portion of this project was done in two parts. Coding for the quantitative analysis of the data was done in Python using VSCode, the second part is the graphical analysis of the data which was done on Jupyter notebook as I had challenges graphing more than one plot in the same VSCode execution.

**LIBRARIES USED:**

**Pandas**: Data manipulation and analysis library.

**NumPy**: Fundamental library for numerical computing.

**Statsmodels**: Estimating and analysing statistical models.

**Scikit-learn (sklearn):** Machine learning library.

**SciPy**: Scientific and technical computing.

**Seaborn**: Data visualization library based on Matplotlib.

The first part I wanted to do is cleaning the data further in Python. As I extracted the data myself, I knew exactly where to look for the missing values. What I chose to do is just delete all rows that contained any null values for any column (as expressed above)

I also want to convert the Sale price column into a float data type as it was stored as a money data type in SQL which placed currency string object in front of the values and commas to separate the 000’s.

The next bit of cleaning I need to do is to remove fields of data not considered important to my model thus I removed the ParcelID, LegalReference, StreetAddress, City and State. For more complex models I could have used binary or hotone encoding however I want to build a linear regression for educational purposes thus I did not consider them useful at this point.

I also need to filter our outliers in my data as this will statistically influence our model. One effective method for detecting and filtering outliers is using standard deviation (or the z-statistic) to filter out data 3 standard deviations away from the mean. However, I chose to use quantile filtering to filter out the top and bottom 1% of all the data for each column.