**Huynh Do Lab#2 Part A:**Objective: This lab assignment exposes student to working with null values, which is data that is missing or is not known within a data set.



1. **Import libraries**

****
The code above imports several Python libraries commonly used in data analysis.

1. **import pandas as pd**
	* A library for data manipulation and analysis (e.g., working with tables, CSVs, etc.).
2. **import numpy as np :**
	* Support for numerical operations and arrays.
3. **from matplotlib import pyplot as plt**
	* Creating static, interactive, and animated plots.
4. **import seaborn as sns**
	* Imports **Seaborn** provides prettier and more informative plots.
5. **from google.colab import files**
	* Used in **Google Colab** to upload/download files from your local system within a Colab notebook.
6. **Upload file NBA.csv**



The above screen shot is used to upload and load a CSV file named **NBA.csv** into a Pandas Data Frame in a Google Colab environment.

When upload is done:


1. Process Null Values
* **Step 1**: Explore null values



* df.isnull() creates a DataFrame of the same shape as df, where each cell is True if it's null, otherwise False.
* .sum() adds up all the True values in each column (since True = 1), giving us the number of missing values per column.
* **Step 2**: Fill with Null values


The code above fills missing (NaN) values in a DataFrame (df) by assigning different fill values depending on each column's data type:

* If a column is numeric (int64 or float64), it fills nulls with 0
* Otherwise (usually text, i.e., object), it fills nulls with "Unknown"

In summary: df\_filled is a copy of the original df, but with all NaN values replaced:

Numbers → 0 | Strings → "Unknown"

**df.fillna({...}) :** Uses that dictionary to fill null values column by column, using the specified defaults.

* **Step 3**: check if all null values are gone



This gives us a summary of the DataFrame after 've filled the missing values. Specifically, it shows:

1. Number of entries (rows)
2. Column names
3. Non-null count for each column — want these to match the total row count if all nulls are gone
4. Data types of each column (e.g., int64, float64, object)
5. Memory usage

The below is result of calling **df\_filled.info()** function



