## Importing Libraries:

In [67]:
```
import pandas as pd
import numpy as np
```

## Importing Data:

In [68]:
```
data_fifa = pd.read_csv("C:/Users/amade/OneDrive/Área de Trabalho/FELIPE(NÃO MEXER)/Projects Data Analisys Portfolio/Data Cleaning and Transformation Pr
display(data_fifa)
```

| | photoUrl | LongName | playerUrl | Nationality | Positions | Name | Age | ↓OVA | POT | Team & Contract | ... | A/W |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | https://cdn.sofifa.com/players/158/023/21_60.png | Lionel Messi | http://sofifa.com/player/158023/lionel-messi/2... | Argentina | RW ST CF | L. Messi | 33 | 93 | 93 | \n\n\n\nFC Barcelona\n2004 ~ 2021\n\n | ... | Medium |
| **1** | https://cdn.sofifa.com/players/020/801/21_60.png | C. Ronaldo dos Santos Aveiro | http://sofifa.com/player/20801/c-ronaldo-dos-s... | Portugal | ST LW | Cristiano Ronaldo | 35 | 92 | 92 | \n\n\n\nJuventus\n2018 ~ 2022\n\n | ... | High |
| **2** | https://cdn.sofifa.com/players/200/389/21_60.png | Jan Oblak | http://sofifa.com/player/200389/jan-oblak/210005/ | Slovenia | GK | J. Oblak | 27 | 91 | 93 | \n\n\n\nAtlético Madrid\n2014 ~ 2023\n\n | ... | Medium |
| **3** | https://cdn.sofifa.com/players/192/985/21_60.png | Kevin De Bruyne | http://sofifa.com/player/192985/kevin-de-bruyn... | Belgium | CAM CM | K. De Bruyne | 29 | 91 | 91 | \n\n\n\nManchester City\n2015 ~ 2023\n\n | ... | High |
| **4** | https://cdn.sofifa.com/players/190/871/21_60.png | Neymar da Silva Santos Jr. | http://sofifa.com/player/190871/neymar-da-silv... | Brazil | LW CAM | Neymar Jr | 28 | 91 | 91 | \n\n\n\nParis Saint-Germain\n2017 ~ 2022\n\n | ... | High |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **18974** | https://cdn.sofifa.com/players/257/710/21_60.png | Mengxuan Zhang | http://sofifa.com/player/257710/mengxuan-zhang... | China PR | CB | Zhang Mengxuan | 21 | 47 | 52 | \n\n\n\nChongqing Dangdai Lifan FC SWM Team\n2... | ... | Low |
| **18975** | https://cdn.sofifa.com/players/258/736/21_60.png | Vani Da Silva | http://sofifa.com/player/258736/vani-da-silva/... | England | ST | V. Da Silva | 17 | 47 | 67 | \n\n\n\nOldham Athletic\n2020 ~ 2021\n\n | ... | Medium |
| **18976** | https://cdn.sofifa.com/players/247/223/21_60.png | Ao Xia | http://sofifa.com/player/247223/ao-xia/210005/ | China PR | CB | Xia Ao | 21 | 47 | 55 | \n\n\n\nWuhan Zall\n2018 ~ 2022\n\n | ... | Medium |
| **18977** | https://cdn.sofifa.com/players/258/760/21_60.png | Ben Hough | http://sofifa.com/player/258760/ben-hough/210005/ | England | CM | B. Hough | 17 | 47 | 67 | \n\n\n\nOldham Athletic\n2020 ~ 2021\n\n | ... | Medium |
| **18978** | https://cdn.sofifa.com/players/255/958/21_60.png | Mateo Flores | http://sofifa.com/player/255958/mateo-flores/2... | Bolivia | CDM | M. Flores | 19 | 47 | 63 | \n\n\n\nClub Bolívar\n2020 ~ 2024\n\n | ... | Medium |

18979 rows × 77 columns

## Analysing Dataset:

In [69]:
```
# ANALYSING COLUMNS:

data_fifa.columns
```

Out[69]:
```
Index(['photoUrl', 'LongName', 'playerUrl', 'Nationality', 'Positions', 'Name',
       'Age', '↓OVA', 'POT', 'Team & Contract', 'ID', 'Height', 'Weight',
       'foot', 'BOV', 'BP', 'Growth', 'Joined', 'Loan Date End', 'Value',
       'Wage', 'Release Clause', 'Attacking', 'Crossing', 'Finishing',
       'Heading Accuracy', 'Short Passing', 'Volleys', 'Skill', 'Dribbling',
       'Curve', 'FK Accuracy', 'Long Passing', 'Ball Control', 'Movement',
       'Acceleration', 'Sprint Speed', 'Agility', 'Reactions', 'Balance',
       'Power', 'Shot Power', 'Jumping', 'Stamina', 'Strength', 'Long Shots',
       'Mentality', 'Aggression', 'Interceptions', 'Positioning', 'Vision',
       'Penalties', 'Composure', 'Defending', 'Marking', 'Standing Tackle',
       'Sliding Tackle', 'Goalkeeping', 'GK Diving', 'GK Handling',
       'GK Kicking', 'GK Positioning', 'GK Reflexes', 'Total Stats',
       'Base Stats', 'W/F', 'SM', 'A/W', 'D/W', 'IR', 'PAC', 'SHO', 'PAS',
       'DRI', 'DEF', 'PHY', 'Hits'],
      dtype='object')
```

In [70]:
```
# ANALYSING DATA INFO:

data_fifa.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18979 entries, 0 to 18978
Data columns (total 77 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   photoUrl          18979 non-null  object
 1   LongName          18979 non-null  object
 2   playerUrl         18979 non-null  object
 3   Nationality       18979 non-null  object
 4   Positions         18979 non-null  object
 5   Name              18979 non-null  object
 6   Age               18979 non-null  int64 
 7   ↓OVA              18979 non-null  int64 
 8   POT               18979 non-null  int64 
 9   Team & Contract   18979 non-null  object
 10  ID                18979 non-null  int64 
 11  Height            18979 non-null  object
 12  Weight            18979 non-null  object
 13  foot              18979 non-null  object
 14  BOV               18979 non-null  int64 
 15  BP                18979 non-null  object
 16  Growth            18979 non-null  int64 
 17  Joined            18979 non-null  object
 18  Loan Date End     1013 non-null   object
 19  Value             18979 non-null  object
 20  Wage              18979 non-null  object
 21  Release Clause    18979 non-null  object
 22  Attacking         18979 non-null  int64 
 23  Crossing          18979 non-null  int64 
 24  Finishing         18979 non-null  int64 
 25  Heading Accuracy  18979 non-null  int64 
 26  Short Passing     18979 non-null  int64 
 27  Volleys           18979 non-null  int64 
 28  Skill             18979 non-null  int64 
 29  Dribbling         18979 non-null  int64 
 30  Curve             18979 non-null  int64 
 31  FK Accuracy       18979 non-null  int64 
 32  Long Passing      18979 non-null  int64 
 33  Ball Control      18979 non-null  int64 
 34  Movement          18979 non-null  int64 
 35  Acceleration      18979 non-null  int64 
 36  Sprint Speed      18979 non-null  int64 
 37  Agility           18979 non-null  int64 
 38  Reactions         18979 non-null  int64 
 39  Balance           18979 non-null  int64 
 40  Power             18979 non-null  int64 
 41  Shot Power        18979 non-null  int64 
 42  Jumping           18979 non-null  int64 
 43  Stamina           18979 non-null  int64 
 44  Strength          18979 non-null  int64 
 45  Long Shots        18979 non-null  int64 
 46  Mentality         18979 non-null  int64 
 47  Aggression        18979 non-null  int64 
 48  Interceptions     18979 non-null  int64 
 49  Positioning       18979 non-null  int64 
 50  Vision            18979 non-null  int64 
 51  Penalties         18979 non-null  int64 
 52  Composure         18979 non-null  int64 
 53  Defending         18979 non-null  int64 
 54  Marking           18979 non-null  int64 
 55  Standing Tackle   18979 non-null  int64 
 56  Sliding Tackle    18979 non-null  int64 
 57  Goalkeeping       18979 non-null  int64 
 58  GK Diving         18979 non-null  int64 
 59  GK Handling       18979 non-null  int64 
 60  GK Kicking        18979 non-null  int64 
 61  GK Positioning    18979 non-null  int64 
 62  GK Reflexes       18979 non-null  int64 
 63  Total Stats       18979 non-null  int64 
 64  Base Stats        18979 non-null  int64 
 65  W/F               18979 non-null  object
 66  SM                18979 non-null  object
 67  A/W               18979 non-null  object
 68  D/W               18979 non-null  object
 69  IR                18979 non-null  object
 70  PAC               18979 non-null  int64 
 71  SHO               18979 non-null  int64 
 72  PAS               18979 non-null  int64 
 73  DRI               18979 non-null  int64 
 74  DEF               18979 non-null  int64 
 75  PHY               18979 non-null  int64 
 76  Hits              18979 non-null  object
dtypes: int64(55), object(22)
memory usage: 11.1+ MB
```

In [71]:
```
# Columns we have to change/drop: Height, Weight, Team & Contract, Joined, Loan Date End, Value, Wage, Release Clause.
```

In [72]:
```
data_fifa[["Height", "Weight", "Team & Contract", "Joined", "Loan Date End", "Value", "Wage", "Release Clause","Hits"]]
```

Out[72]:

| | Height | Weight | Team & Contract | Joined | Loan Date End | Value | Wage | Release Clause | Hits |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 5'7" | 159lbs | \n\n\n\nFC Barcelona\n2004 ~ 2021\n\n | Jul 1, 2004 | NaN | €67.5M | €560K | €138.4M | \n372 |
| **1** | 6'2" | 183lbs | \n\n\n\nJuventus\n2018 ~ 2022\n\n | Jul 10, 2018 | NaN | €46M | €220K | €75.9M | \n344 |
| **2** | 6'2" | 192lbs | \n\n\n\nAtlético Madrid\n2014 ~ 2023\n\n | Jul 16, 2014 | NaN | €75M | €125K | €159.4M | \n86 |
| **3** | 5'11" | 154lbs | \n\n\n\nManchester City\n2015 ~ 2023\n\n | Aug 30, 2015 | NaN | €87M | €370K | €161M | \n163 |
| **4** | 5'9" | 150lbs | \n\n\n\nParis Saint-Germain\n2017 ~ 2022\n\n | Aug 3, 2017 | NaN | €90M | €270K | €166.5M | \n273 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **18974** | 5'10" | 154lbs | \n\n\n\nChongqing Dangdai Lifan FC SWM Team\n2... | Aug 1, 2020 | NaN | €35K | €1K | €57K | \n2 |
| **18975** | 5'7" | 128lbs | \n\n\n\nOldham Athletic\n2020 ~ 2021\n\n | Aug 1, 2020 | NaN | €60K | €500 | €165K | \n3 |
| **18976** | 5'10" | 146lbs | \n\n\n\nWuhan Zall\n2018 ~ 2022\n\n | Jul 13, 2018 | NaN | €40K | €1K | €70K | \n3 |
| **18977** | 5'9" | 143lbs | \n\n\n\nOldham Athletic\n2020 ~ 2021\n\n | Aug 1, 2020 | NaN | €60K | €500 | €165K | \n5 |
| **18978** | 5'9" | 150lbs | \n\n\n\nClub Bolívar\n2020 ~ 2024\n\n | Jan 1, 2020 | NaN | €60K | €500 | €167K | \n2 |

18979 rows × 9 columns

In [73]:
```
# (DATA CLEANING AND TRANSFORMATION PROCESS):

# 1º Drop columns(photoUrl, playerUrl and Loan Date End).
# 2º Rename colum Long Name to Full Name, OVA to Overall and POT to Potential.
# 3º Trasform Height from feet to cm + convert dtype to int.
# 4º Transform Weight from pounds to kg + convert dtype to int.
# 5º Transform Joined column from : Month day, year, to : month-day-year.
# 6º Split Team and contract columns and remove the \n´s: \n\n\n\nFC Barcelona\n2004 ~ 2021\n\n ,to : FC Barcelona / 2004 ~ 2021.
# 7º Create a column years with club (end contract - start contract).
# 8° Transform Value,Wage and Release Clause collumns into normal extendend numbers without currency symbols  + convert dtype to int.
# 9° Remove the \n's from hits and convert dtype to int.
# 10° Remove the star symbol(★) from the columns that have it.
```

## Transforming Data:

In [74]:
```
# (DROPING COLUMNS PHOTOURL, PLAYERURL AND LOAN DATE END):

data_fifa = data_fifa.drop(["photoUrl", "playerUrl", "Loan Date End"], axis = 1)
data_fifa.shape
```

Out[74]:
```
(18979, 74)
```

```
In [75]: # (RENAMING COLUMN LONG NAME TO FULL NAME, OVA TO OVERALL AND POT TO POTENTIAL):

         data_fifa = data_fifa.rename(columns={'LongName': 'Full Name', '↓OVA': 'Overall', 'POT': 'Potential'})
         display(data_fifa[['Full Name','Overall' ,'Potential']])
```

| | Full Name | Overall | Potential |
|---|---|---|---|
| **0** | Lionel Messi | 93 | 93 |
| **1** | C. Ronaldo dos Santos Aveiro | 92 | 92 |
| **2** | Jan Oblak | 91 | 93 |
| **3** | Kevin De Bruyne | 91 | 91 |
| **4** | Neymar da Silva Santos Jr. | 91 | 91 |
| **...** | ... | ... | ... |
| **18974** | Mengxuan Zhang | 47 | 52 |
| **18975** | Vani Da Silva | 47 | 67 |
| **18976** | Ao Xia | 47 | 55 |
| **18977** | Ben Hough | 47 | 67 |
| **18978** | Mateo Flores | 47 | 63 |

18979 rows × 3 columns

```
In [76]: # (1º TRASFORMING HEIGHT FROM FEET TO CM + CONVERTING DTYPE TO INT):

         data_fifa["Height"] = data_fifa["Height"].str.replace("'", ".") # replacing ' for .
```

```
In [77]: data_fifa["Height"] = data_fifa["Height"].apply(lambda x: x.replace('"', '')) # replacing " for empty
         data_fifa["Height"]
```

```
Out[77]: 0         5.7
         1         6.2
         2         6.2
         3        5.11
         4         5.9
                  ...
         18974    5.10
         18975     5.7
         18976    5.10
         18977     5.9
         18978     5.9
         Name: Height, Length: 18979, dtype: object
```

In [78]:
```
data_fifa["Height"] = data_fifa["Height"].astype(str).astype(float) # Transforming from Object to float, to make possible the
# multiplication from feet to cm with a higher precision.
```

In [79]:
```
data_fifa["Height"] = data_fifa["Height"].apply(lambda x: x * 30.48) # Transforming height from feet to cm

data_fifa["Height"] = data_fifa["Height"].astype(int) # Transforming column from float to int.
data_fifa["Height"]
```

Out[79]:
```
0        173
1        188
2        188
3        155
4        179
        ...
18974    155
18975    173
18976    155
18977    179
18978    179
Name: Height, Length: 18979, dtype: int32
```

In [80]:
```
# (2º TRANSFORMING WEIGHT FROM POUNDS TO KG + CONVERTING DTYPE TO INT):
```

In [81]:
```
data_fifa["Weight"] = data_fifa["Weight"].apply(lambda x: x.replace("lbs", "")) # Removing the lbs from the weight value.
data_fifa["Weight"]
```

Out[81]:
```
0        159
1        183
2        192
3        154
4        150
        ...
18974    154
18975    128
18976    146
18977    143
18978    150
Name: Weight, Length: 18979, dtype: object
```

In [82]:
```
data_fifa["Weight"] = data_fifa["Weight"].astype(str).astype(float)# transforming column weight from object type to float to
# make possible the multiplication from pounds to kg with a higher precision.
data_fifa["Weight"]
```

Out[82]:
```
0        159.0
1        183.0
2        192.0
3        154.0
4        150.0
         ...
18974    154.0
18975    128.0
18976    146.0
18977    143.0
18978    150.0
Name: Weight, Length: 18979, dtype: float64
```

In [83]:
```
data_fifa["Weight"] = data_fifa["Weight"].apply(lambda x: x * 0.453) # Transforming weight from pounds to Kg
data_fifa["Weight"]
```

Out[83]:
```
0        72.027
1        82.899
2        86.976
3        69.762
4        67.950
          ...
18974    69.762
18975    57.984
18976    66.138
18977    64.779
18978    67.950
Name: Weight, Length: 18979, dtype: float64
```

In [84]:
```
data_fifa["Weight"] = data_fifa["Weight"].astype(int) # Transforming column from float to int.
data_fifa["Weight"]
```

Out[84]:
```
0        72
1        82
2        86
3        69
4        67
         ..
18974    69
18975    57
18976    66
18977    64
18978    67
Name: Weight, Length: 18979, dtype: int32
```

In [85]:
```
# (3º JOINED FROM : MONTH DAY, YEAR, TO : MONTH-DAY-YEAR):
```

In [86]:
```
data_fifa["Joined"]
```

Out[86]:
```
0         Jul 1, 2004
1        Jul 10, 2018
2        Jul 16, 2014
3        Aug 30, 2015
4         Aug 3, 2017
             ...
18974     Aug 1, 2020
18975     Aug 1, 2020
18976    Jul 13, 2018
18977     Aug 1, 2020
18978     Jan 1, 2020
Name: Joined, Length: 18979, dtype: object
```

In [87]:
```
def transform_date(df, column_name): # Function that transform Jul 1, 2004 to 07-01-2004.
    # Convert the column to datetime format
    data_fifa["Joined"] = pd.to_datetime(data_fifa["Joined"], format="%b %d, %Y")

    # Transform the date format to MONTH-DAY-YEAR
    data_fifa["Joined"] = data_fifa["Joined"].dt.strftime("%m-%d-%Y")

    return df

data_fifa = transform_date(data_fifa, 'Joined')
```

In [88]:
```
data_fifa["Joined"]
```

Out[88]:
```
0        07-01-2004
1        07-10-2018
2        07-16-2014
3        08-30-2015
4        08-03-2017
            ...
18974    08-01-2020
18975    08-01-2020
18976    07-13-2018
18977    08-01-2020
18978    01-01-2020
Name: Joined, Length: 18979, dtype: object
```

In [89]:
```
# (4º Transforming Weight from pounds to kg + convert dtype to int):
```

In [90]:
```
data_fifa['Team & Contract'] = data_fifa['Team & Contract'].astype('str') # Creating a function that splits Team from Contract.
data_fifa['Team & Contract'].replace('\n', '', regex=True, inplace=True)
test = data_fifa['Team & Contract'][0]
Team = []
Contract_Duration = []
for x in range(len(data_fifa['Team & Contract'])):
    Team.append(str(data_fifa['Team & Contract'][x][:-11]))
    c = str(data_fifa['Team & Contract'][x][-11:])
    if c.startswith("2") == True:
        Contract_Duration.append(c)
    else:
        Contract_Duration.append("0")
data_fifa = data_fifa.drop(columns = ['Team & Contract'])
data_fifa['Team'] = Team
data_fifa['Contract Duration'] = Contract_Duration
```

In [91]:
```
data_fifa[["Team", "Contract Duration"]]
```

Out[91]:

| | Team | Contract Duration |
|---|---|---|
| **0** | FC Barcelona | 2004 ~ 2021 |
| **1** | Juventus | 2018 ~ 2022 |
| **2** | Atlético Madrid | 2014 ~ 2023 |
| **3** | Manchester City | 2015 ~ 2023 |
| **4** | Paris Saint-Germain | 2017 ~ 2022 |
| **...** | ... | ... |
| **18974** | Chongqing Dangdai Lifan FC SWM Team | 2020 ~ 2020 |
| **18975** | Oldham Athletic | 2020 ~ 2021 |
| **18976** | Wuhan Zall | 2018 ~ 2022 |
| **18977** | Oldham Athletic | 2020 ~ 2021 |
| **18978** | Club Bolívar | 2020 ~ 2024 |

18979 rows × 2 columns

In [92]:
```
# (5º CREATING A COLUMN YEARS WITH CLUB):
```

In [93]:
```
def subtract_values(column): # Function that subtracts end of contract from start of contract.
    # Split the values in the column based on "~"
    values = column.str.split("~")

    # Apply the subtraction operation and set the result to 0 if it's already 0
    subtracted_values = values.apply(lambda x: float(x[1]) - float(x[0]) if x[0] != '0' else 0)

    return subtracted_values
```

In [94]:
```
data_fifa['Years_in_Club'] = subtract_values(data_fifa["Contract Duration"]) # Creating new column Years_in_Club.
```

In [95]:
```
data_fifa['Years_in_Club'] = data_fifa['Years_in_Club'].astype(int)
```

In [96]:
```
# (6º SPLIT TEAM AND CONTRACT COLUMNS AND REMOVE THE \n´s: \n\n\n\nFC Barcelona\n2004 ~ 2021\n\n ,to : FC Barcelona / 2004 ~ 2021)
```

In [97]:
```
data_fifa[["Value","Wage","Release Clause"]]
```

Out[97]:

| | Value | Wage | Release Clause |
|---|---|---|---|
| **0** | €67.5M | €560K | €138.4M |
| **1** | €46M | €220K | €75.9M |
| **2** | €75M | €125K | €159.4M |
| **3** | €87M | €370K | €161M |
| **4** | €90M | €270K | €166.5M |
| **...** | ... | ... | ... |
| **18974** | €35K | €1K | €57K |
| **18975** | €60K | €500 | €165K |
| **18976** | €40K | €1K | €70K |
| **18977** | €60K | €500 | €165K |
| **18978** | €60K | €500 | €167K |

18979 rows × 3 columns

In [98]:
```
data_fifa["Value"] = data_fifa["Value"].apply(lambda x: x.replace("€", "")) # Removing the euro sign(€) from the values
data_fifa["Wage"] = data_fifa["Wage"].apply(lambda x: x.replace("€", ""))
data_fifa["Release Clause"] = data_fifa["Release Clause"].apply(lambda x: x.replace("€", ""))
```

In [99]:
```
# Function to convert the letter to a multiplier
def get_multiplier(letter):
    if letter == 'M':
        return 1000000
    elif letter == 'K':
        return 1000
    else:
        return 1


# Iterate over each value in the column and update it
for index, value in data_fifa["Value"].items():
    if value[-1].isdigit():
        # No letter present, so no multiplication needed
        continue
    multiplier = get_multiplier(value[-1])
    number = float(value[:-1])
    data_fifa.at[index, "Value"] = number * multiplier
```

In [100]:
```
# Iterate over each value in the column and update it
for index, value in data_fifa["Release Clause"].items():
    if value[-1].isdigit():
        # No letter present, so no multiplication needed
        continue
    multiplier = get_multiplier(value[-1])
    number = float(value[:-1])
    data_fifa.at[index, "Release Clause"] = number * multiplier
```

In [101]:
```
# Iterate over each value in the column and update it
for index, value in data_fifa["Wage"].items():
    if value[-1].isdigit():
        # No letter present, so no multiplication needed
        continue
    multiplier = get_multiplier(value[-1])
    number = float(value[:-1])
    data_fifa.at[index, "Wage"] = number * multiplier
```

In [102]:
```
data_fifa[["Value","Wage","Release Clause"]]
```

Out[102]:

| | Value | Wage | Release Clause |
|---|---|---|---|
| **0** | 67500000.0 | 560000.0 | 138400000.0 |
| **1** | 46000000.0 | 220000.0 | 75900000.0 |
| **2** | 75000000.0 | 125000.0 | 159400000.0 |
| **3** | 87000000.0 | 370000.0 | 161000000.0 |
| **4** | 90000000.0 | 270000.0 | 166500000.0 |
| **...** | ... | ... | ... |
| **18974** | 35000.0 | 1000.0 | 57000.0 |
| **18975** | 60000.0 | 500 | 165000.0 |
| **18976** | 40000.0 | 1000.0 | 70000.0 |
| **18977** | 60000.0 | 500 | 165000.0 |
| **18978** | 60000.0 | 500 | 167000.0 |

18979 rows × 3 columns

In [103]:
```
# Transforming dtype from object to float !!

data_fifa[["Value","Wage","Release Clause"]] = data_fifa[["Value","Wage","Release Clause"]].astype(str).astype(float)
```

In [104]:
```
data_fifa[["Value","Wage","Release Clause"]].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18979 entries, 0 to 18978
Data columns (total 3 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Value           18979 non-null  float64
 1   Wage            18979 non-null  float64
 2   Release Clause  18979 non-null  float64
dtypes: float64(3)
memory usage: 444.9 KB
```

In [105]:
```
# (7° REMOVING THE \n'S FROM HITS + CONVERT DTYPE TO INT):
```

In [106]:
```
# Visualizing data before transformation:
data_fifa["Hits"].info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 18979 entries, 0 to 18978
Series name: Hits
Non-Null Count  Dtype
--------------  -----
18979 non-null  object
dtypes: object(1)
memory usage: 148.4+ KB
```

In [107]:
```
data_fifa["Hits"].describe()
```

Out[107]:
```
count     18979
unique      374
top         \n1
freq       4321
Name: Hits, dtype: object
```

In [108]:
```
set(data_fifa["Hits"])
```

Out[108]:
```
{'\n1',
 '\n1.1K',
 '\n1.2K',
 '\n1.3K',
 '\n1.5K',
 '\n1.7K',
 '\n1.8K',
 '\n1.9K',
 '\n10',
 '\n100',
 '\n101',
 '\n102',
 '\n103',
 '\n104',
 '\n105',
 '\n106',
 '\n107',
 '\n108',
 '\n109',
```

In [109]:
```
data_fifa["Hits"] = data_fifa["Hits"].replace(r'\n','', regex=True)# Removing the \n´s from the column.
```

In [110]:
```
# Function to convert the letter to a multiplier
def get_multiplier(letter):
    if letter == 'M':
        return 1000000
    elif letter == 'K':
        return 1000
    else:
        return 1


# Iterate over each value in the column and update it
for index, value in data_fifa["Hits"].items():
    if value[-1].isdigit():
        # No letter present, so no multiplication needed
        continue
    multiplier = get_multiplier(value[-1])
    number = float(value[:-1])
    data_fifa.at[index, "Hits"] = number * multiplier
```

In [111]:
```
data_fifa["Hits"] = data_fifa["Hits"].astype(int) # Converting data from object to int.
```

In [112]:
```
# Visualizing data after transformation:

data_fifa[["Hits", "Full Name"]]
```

Out[112]:

| | Hits | Full Name |
|---|---|---|
| **0** | 372 | Lionel Messi |
| **1** | 344 | C. Ronaldo dos Santos Aveiro |
| **2** | 86 | Jan Oblak |
| **3** | 163 | Kevin De Bruyne |
| **4** | 273 | Neymar da Silva Santos Jr. |
| **...** | ... | ... |
| **18974** | 2 | Mengxuan Zhang |
| **18975** | 3 | Vani Da Silva |
| **18976** | 3 | Ao Xia |
| **18977** | 5 | Ben Hough |
| **18978** | 2 | Mateo Flores |

18979 rows × 2 columns

In [113]:
```
set(data_fifa["Hits"])
```

Out[113]:
```
{1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
 10,
 11,
 12,
 13,
 14,
 15,
 16,
 17,
 18,
 19,
```

In [114]:
```
data_fifa["Hits"].info()
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 18979 entries, 0 to 18978
Series name: Hits
Non-Null Count  Dtype
--------------  -----
18979 non-null  int32
dtypes: int32(1)
memory usage: 74.3 KB
```

In [115]:
```
data_fifa["Hits"].describe()
```

Out[115]:
```
count    18979.000000
mean        16.934085
std         73.203131
min          1.000000
25%          2.000000
50%          3.000000
75%         10.000000
max       4500.000000
Name: Hits, dtype: float64
```

In [116]:
```
# (8° Remove the star symbol(★) from the columns W/F, SM and IR):
```

In [117]:
```
# Search for columns that have the star symbol.

# Specify the character to search for
character_to_find = '★'

# Iterate over the columns and check for the character
columns_with_character = []
for column in data_fifa.columns:
    if any(data_fifa[column].astype(str).str.contains(character_to_find)):
        columns_with_character.append(column)

print(f"The columns that contain the character '{character_to_find}' are: {columns_with_character}")
```

```
The columns that contain the character '★' are: ['W/F', 'SM', 'IR']
```

In [118]:
```
data_fifa[["W/F", "SM", "IR"]]
```

Out[118]:

| | W/F | SM | IR |
|---|---|---|---|
| **0** | 4 ★ | 4★ | 5 ★ |
| **1** | 4 ★ | 5★ | 5 ★ |
| **2** | 3 ★ | 1★ | 3 ★ |
| **3** | 5 ★ | 4★ | 4 ★ |
| **4** | 5 ★ | 5★ | 5 ★ |
| **...** | ... | ... | ... |
| **18974** | 2 ★ | 2★ | 1 ★ |
| **18975** | 2 ★ | 2★ | 1 ★ |
| **18976** | 2 ★ | 2★ | 1 ★ |
| **18977** | 2 ★ | 2★ | 1 ★ |
| **18978** | 3 ★ | 2★ | 1 ★ |

18979 rows × 3 columns

In [119]:
```
data_fifa[["W/F", "SM", "IR"]] = data_fifa[["W/F", "SM", "IR"]].apply(lambda x: x.str.replace('★', '')) # removing star from columns that have it !!!.
```

```
In [120]: data_fifa[["W/F", "SM", "IR"]]
```

```
Out[120]:
        W/F  SM  IR
     0    4   4   5
     1    4   5   5
     2    3   1   3
     3    5   4   4
     4    5   5   5
   ...  ...  ... ...
 18974    2   2   1
 18975    2   2   1
 18976    2   2   1
 18977    2   2   1
 18978    3   2   1

18979 rows × 3 columns
```

```
In [ ]:
```