In [1]:
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [3]:
```python
#Loading the Dataset for analysis
data=pd.read_csv('fifa.csv')
data.head()
```

```
C:\Users\WINDOWS10\AppData\Local\Temp\ipykernel_9620\613872930.py:1: DtypeWarning: Columns (76) have mixed types. Specify dtype option on import or set low_memory=False.
  data=pd.read_csv('fifa.csv')
```

Out[3]:

	photoUrl	LongName	playerUrl	Nationality	Positions	Name	Age	↓OVA	POT	Team & Contract	...	A/W	D/W	IR	PAC	SHO	PAS	DRI	DEF	PH
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	Low	5 ★	85	92	91	95	38	6
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	Low	5 ★	89	93	81	89	35	7
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	Medium	3 ★	87	92	78	90	52	9
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	High	4 ★	76	86	93	88	64	7
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	Medium	5 ★	91	85	86	94	36	5

5 rows × 77 columns

In [5]:
```python
#Loading the dataset into pandas dataframe
df=pd.DataFrame(data)
df.head()
```

Out[5]:

	photoUrl	LongName	playerUrl	Nationality	Positions	Name	Age	↓OVA	POT	Team & Contract	...	A/W	D/W	IR	PAC	SHO	PAS	DRI	DEF	PH
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	Low	5 ★	85	92	91	95	38	6
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	Low	5 ★	89	93	81	89	35	7
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	Medium	3 ★	87	92	78	90	52	9
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	High	4 ★	76	86	93	88	64	7
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	Medium	5 ★	91	85	86	94	36	5

5 rows × 77 columns

In [6]:
```python
#Checking what are the columns has in this dataset
df.columns
```

Out[6]:
```
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 [9]:
```python
#Deleting columns which is not necessary in the dataset
df.drop(columns=['photoUrl','playerUrl','Team & Contract'],inplace=True)
```

In [11]:
```python
#Calculating total raws and total columns
df.shape
```

Out[11]:
```
(18979, 74)
```

In [13]:
```python
#Calculating total null values in the dataset
df.isnull().sum().sum()
```

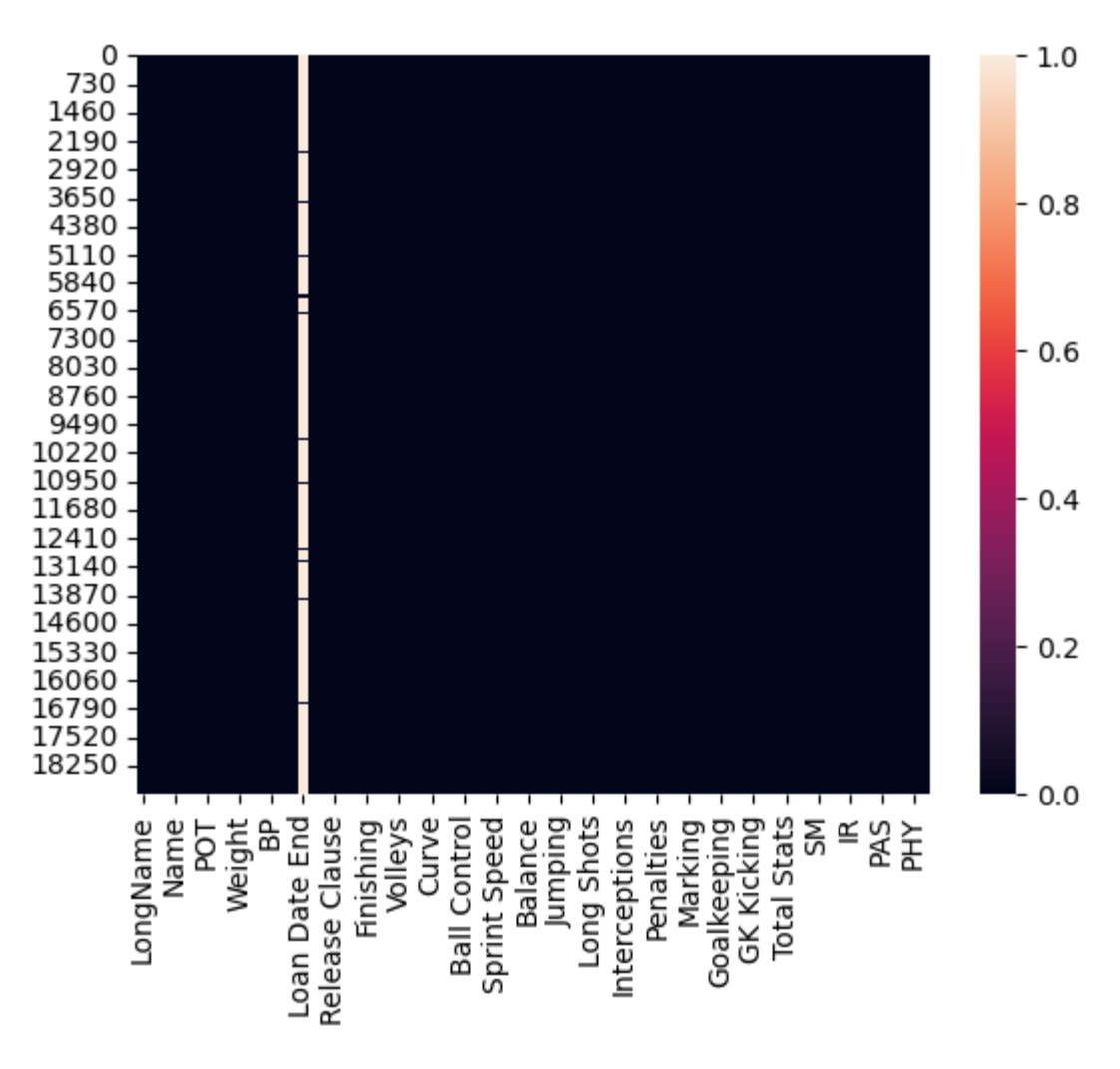
Out[13]:
```
17966
```

In [20]:
```python
#Calculating what is percentage of total null value comparing to tatal value in the dataset
((df.isnull().sum().sum())/(df.shape[0]*df.shape[1]))*100
```

Out[20]:
```
1.2792232666830907
```

In [28]:
```python
#Visualizing null values through heat map,it is the best way to see the null values in the dataset
sns.heatmap(df.isnull())
```

Out[28]:
```
<Axes: >
```



In [36]:
```python
#from the above heat map we found that only "Loan Date End" column has the null value which is 94% which is now irrelevance
#column because of more null value is present which is not desirable
(df['Loan Date End'].isnull().sum())/(df.shape[0])*100
```

Out[36]:
```
94.6625217345487
```

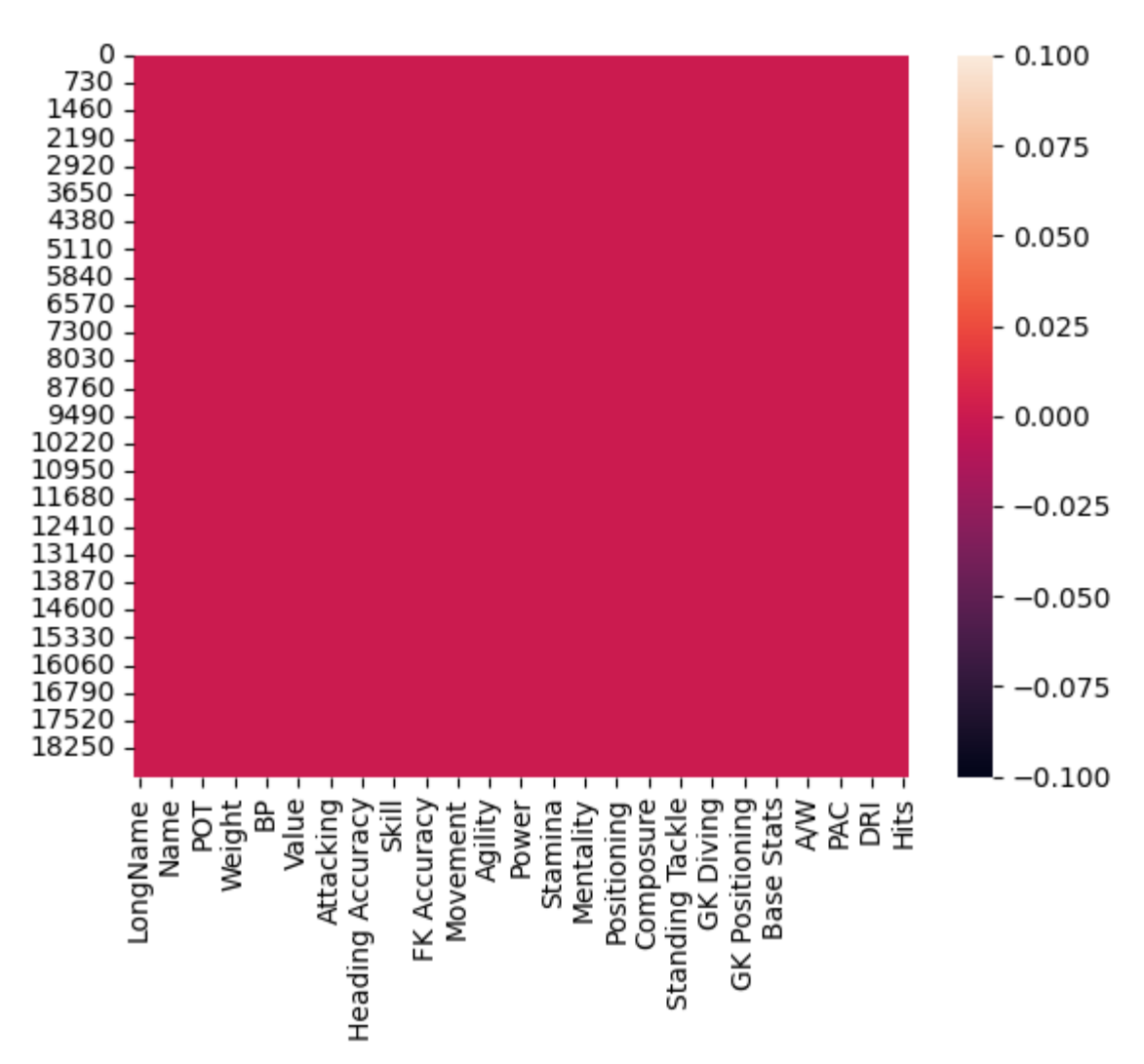
In [37]:
```python
#dropping the "Loan Date End " column
df.drop(columns=['Loan Date End'],inplace=True)
```

In [39]:
```python
#now calculating the null values in the dataset which is 'zero' now
df.isnull().sum().sum()
```

Out[39]:
```
0
```

In [41]:
```python
#We can visualize this through heat map also
sns.heatmap(df.isnull())
```

Out[41]:
```
<Axes: >
```



In [48]:
```python
#Now I want to know what are the data types my dataset has
df.dtypes.unique()
```

Out[48]:
```
array([dtype('O'), dtype('int64')], dtype=object)
```

In [163…
```python
#From the above we knew that this data set has two types of dataset which is 'object'and 'int64' data types
#Acessing the 'Object' types columns
object_type=df.select_dtypes("object").columns
for i in object_type:
    print(i)
```

```
LongName
Nationality
Positions
Name
Height
Weight
foot
BP
Value
Wage
Release Clause
W/F
SM
A/W
D/W
IR
Hits
```

In [162…
```python
#Accessing "int64" data types columns
int_type=df.select_dtypes("int64").columns
for i in int_type:
    print(i)
```

```
Age
↓OVA
POT
ID
BOV
Growth
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
PAC
SHO
PAS
DRI
DEF
PHY
Year
```

In [53]:
```python
#from the above we come to know that column 'Joined' has object data types whcih is a date column so we need to change it
#date data type format
df['Joined']=pd.to_datetime(df['Joined'])
```

In [56]:
```python
#Checking that 'Joined' columns data types change or not to date format
df['Joined'].dtypes
```

Out[56]:
```
dtype('<M8[ns]')
```

In [58]:
```python
df['Year']=df['Joined'].dt.year
```

In [68]:
```python
df['ID'].duplicated().value_counts()
```

Out[68]:
```
False    18978
True         1
Name: ID, dtype: int64
```

In [70]:
```python
df['ID'].drop_duplicates(inplace=True)
```

In [161…
```python
for i in df['Nationality'].unique():
    print(i)
```

```
Argentina
Portugal
Slovenia
Belgium
Brazil
Poland
France
Egypt
Senegal
Netherlands
Germany
Spain
England
Scotland
Korea Republic
Costa Rica
Italy
Gabon
Croatia
Uruguay
Switzerland
Slovakia
Serbia
Morocco
Algeria
Denmark
Hungary
Bosnia Herzegovina
Norway
Nigeria
Cameroon
Ghana
Mexico
Austria
Colombia
Albania
Chile
Ivory Coast
Greece
Finland
Wales
Sweden
Czech Republic
Togo
Russia
Canada
United States
Guinea
Venezuela
Montenegro
Israel
Republic of Ireland
Ukraine
Ecuador
Turkey
Jamaica
Australia
DR Congo
Armenia
China PR
Northern Ireland
North Macedonia
Kosovo
Mali
Peru
Central African Republic
Iceland
Burkina Faso
Paraguay
Japan
Romania
New Zealand
Angola
Tunisia
Iran
Syria
Dominican Republic
Cape Verde
Kenya
Georgia
Zambia
Panama
Equatorial Guinea
Tanzania
Zimbabwe
Congo
Moldova
South Africa
Guinea Bissau
Mozambique
Honduras
Iraq
Cuba
Cyprus
Lithuania
Estonia
Madagascar
Benin
Curacao
Saudi Arabia
Gambia
Uzbekistan
Chad
Libya
Philippines
Sierra Leone
Liberia
Bulgaria
Comoros
Saint Kitts and Nevis
United Arab Emirates
Namibia
Luxembourg
Trinidad & Tobago
Bermuda
Thailand
Burundi
New Caledonia
Puerto Rico
Bolivia
Kazakhstan
Antigua & Barbuda
Latvia
Malawi
Montserrat
São Tomé & Príncipe
Mauritania
Jordan
Eritrea
El Salvador
Aruba
Uganda
Chinese Taipei
Azerbaijan
Afghanistan
Faroe Islands
Haiti
Sudan
Grenada
Lebanon
Guam
Palestine
Belarus
Guyana
Rwanda
Saint Lucia
Papua New Guinea
Liechtenstein
India
Ethiopia
Belize
Andorra
Guatemala
Malta
Niger
Korea DPR
Barbados
Macau
South Sudan
Singapore
Hong Kong
Nicaragua
Malaysia
Indonesia
```

In [160…
```python
#Now I want to know in which year maximum people joined
df.groupby(['Year']).agg({'ID':'count'}).nlargest(5,'ID')
```

Out[160]:

Year	ID
2020	5868
2019	5623
2018	3063
2017	1795
2016	972

From the above we came to know that in the year 2020 the maximum number of people joined which is 5868

In [147…
```python
#Now I want to know which 10 country has most number of the player
df.groupby(['Nationality']).agg({'ID':'count'}).nlargest(10,'ID')
```

Out[147]:

Nationality	ID
England	1704
Germany	1195
Spain	1065
France	1003
Argentina	943
Brazil	887
Japan	485
Netherlands	438
Italy	387
Sweden	380

In [157…
```python
#Visulazing top 10 nationalites with most number of players through bar plot
df.groupby(['Nationality']).agg({'ID':'count'}).nlargest(10,'ID').plot(kind='bar',color='orange')
plt.xlabel('Nationality')
plt.ylabel('Number of Players')
plt.title('Top 10 Nationalities with Most Players')
plt.show()
```

In [171…
```python
df['Wage']=df['Wage'].str.replace('€', '').str.replace('K', 'e3')
```

In [173…
```python
df['Wage']=pd.to_numeric(df['Wage'])
```

In [175…
```python
df['Wage'].dtypes
```

Out[175]:
```
dtype('float64')
```

In [214…
```python
#Finding top 10 players who has maximum Salary or wage
df.nlargest(10,'Wage')[['Name','Wage']]
```

Out[214]:

	Name	Wage
0	L. Messi	560000.0
3	K. De Bruyne	370000.0
15	K. Benzema	350000.0
25	E. Hazard	350000.0
12	Casemiro	310000.0
26	T. Kroos	310000.0
16	Sergio Ramos	300000.0
17	S. Agüero	300000.0
37	A. Griezmann	290000.0
4	Neymar Jr	270000.0

In [222…
```python
df.nlargest(10,'Goalkeeping')[['LongName','Goalkeeping','IR']]
```

Out[222]:

	LongName	Goalkeeping	IR
14	Manuel Neuer	440	5 ★
7	Alisson Ramses Becker	439	3 ★
11	Marc-André ter Stegen	439	3 ★
2	Jan Oblak	437	3 ★
21	Ederson Santana de Moraes	435	2 ★
27	Samir Handanovič	424	3 ★
58	Yann Sommer	423	3 ★
91	Péter Gulácsi	421	2 ★
13	Thibaut Courtois	420	4 ★
104	André Onana	419	3 ★

In [231…
```python
df.nlargest(10,'DEF')[['LongName','DEF','IR']]
```

Out[231]:

	LongName	DEF	IR
10	Virgil van Dijk	91	3 ★
47	Giorgio Chiellini	90	4 ★
24	Kalidou Koulibaly	89	3 ★
57	Mats Hummels	89	4 ★
16	Sergio Ramos García	88	4 ★
32	Aymeric Laporte	88	2 ★
50	Raphaël Varane	87	3 ★
64	Milan Škriniar	87	1 ★
71	Clément Lenglet	87	2 ★
75	Marcos Aoás Corrêa	87	3 ★

In [237…
```python
def convert_to_inches(height_str):
    feet, inches = height_str.split("'")
    total_inches = int(feet) * 12 + int(inches.replace('"', ''))
    return total_inches

# Apply the function to the 'Height' column
df['Height']=df['Height'].apply(convert_to_inches)
```

In [240…
```python
df.Height
```

Out[240]:
```
0        67
1        74
2        74
3        71
4        69
         ..
18974    70
18975    67
18976    70
18977    69
18978    69
Name: Height, Length: 18979, dtype: int64
```

In [245…
```python
#Who is the tallest player
df.nlargest(1,'Height')[['LongName','Height']]
```

Out[245]:

	LongName	Height
10305	Tomáš Holý	81

In [257…
```python
df.nlargest(10,'Penalties')[['LongName','Penalties','IR']]
```

Out[257]:

	LongName	Penalties	IR
4	Neymar da Silva Santos Jr.	92	5 ★
16	Sergio Ramos García	92	4 ★
119	Raúl Jiménez	92	3 ★
354	Max Kruse	92	2 ★
1586	Mark Noble	92	2 ★
33	Bruno Miguel Borges Fernandes	91	2 ★
90	Marco Reus	91	4 ★
172	Luiz Frello Filho Jorge	91	2 ★
322	Sébastien Haller	91	2 ★
23	Harry Kane	90	3 ★