In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
#importing orginal dataset with all results
sports_events=pd.read_csv('/Users/alexandergandji/Desktop/Course Folder/Capstone Project SQL Data Science /Spor
```

In [6]:
```
sports_events.head()
```

Out[6]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN

Nationality

In [7]:
```
#importing orginal dataset
nat=pd.read_csv('/Users/alexandergandji/Desktop/Course Folder/Capstone Project SQL Data Science /Sports/SportsS
```

In [8]:
```
#creating a dataframe for all particpating nations
nat_df1=pd.DataFrame(nat, columns=['NOC'])
nat_df1=nat_df1.append({'NOC':'SGP'}, ignore_index=True)
```

In [9]:
```
#alocating a unique number to every value
nat_df1['NOC_ID']=range(1,232)
```

In [10]:
```
#setting a new index
nat_df1=nat_df1.set_index('NOC_ID')
```

In [11]:
```
nat_df1
```

Out[11]:

	NOC
NOC_ID	
1	AFG
2	AHO
3	ALB
4	ALG
5	AND
...	...
227	YMD
228	YUG
229	ZAM
230	ZIM
231	SGP

231 rows × 1 columns

In [12]:
```
#saving it on a csv file
nat_df1.to_csv('/Users/alexandergandji/Desktop/Course Folder/Capstone Project SQL Data Science /Sports/national
```

Medal

In [17]:
```
sports_events
```

Out[17]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Even
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketba Men Basketba
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men Extr Lightweig
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Footba Men Footba
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-O War Men Tug-O W
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Spee Skatin Women 500 metre
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
271111	135569	Andrzej ya	M	29.0	179.0	89.0	Poland-1	POL	1976 Winter	1976	Winter	Innsbruck	Luge	Luge Mixe (Men) Doubl
271112	135570	Piotr ya	M	27.0	176.0	59.0	Poland	POL	2014 Winter	2014	Winter	Sochi	Ski Jumping	S Jumpi Men Large Hi Individu
271113	135570	Piotr ya	M	27.0	176.0	59.0	Poland	POL	2014 Winter	2014	Winter	Sochi	Ski Jumping	S Jumpi Men Large Hi Tea
271114	135571	Tomasz Ireneusz ya	M	30.0	185.0	96.0	Poland	POL	1998 Winter	1998	Winter	Nagano	Bobsleigh	Bobsleig Men's Fo
271115	135571	Tomasz Ireneusz ya	M	34.0	185.0	96.0	Poland	POL	2002 Winter	2002	Winter	Salt Lake City	Bobsleigh	Bobsleig Men's Fo

271116 rows × 15 columns

In [18]:
```
#creating a new dataframe with the different medal types
medal_df=pd.DataFrame(sports_events.Medal.unique().tolist(), columns=['Medal'])
```

In [19]:
```
#Nan is considered as not winning a medal, I have transformed it to 'No_Medal'
medal_df=medal_df.fillna('No_Medal')
```

In [20]:
```
medal_df
```

Out[20]:

	Medal
0	No_Medal
1	Gold
2	Bronze
3	Silver

In [21]:
```
#using lambda to allocate a number to each medal type
medal_df['Medal_ID']=medal_df['Medal'].apply(lambda x: 1 if x=='Gold' else (2 if x=='Silver' else
                                              (3 if x=='Bronze' else 4)))
```

In [22]:
```
medal_df
```

Out[22]:

	Medal	Medal_ID
0	No_Medal	4
1	Gold	1
2	Bronze	3
3	Silver	2

In [23]:
```
#sorting values
medal_df.sort_values('Medal_ID', inplace=True)
```

In [24]:
```
#setting a new index
medal_df.set_index('Medal_ID', inplace=True)
```

In [25]:
```
#saving it on a csv file
medal_df.to_csv('/Users/alexandergandji/Desktop/Course Folder/Capstone Project SQL Data Science /Sports/olympic
```

olympic games

In [26]:
```
sports_events.head()
```

Out[26]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN

In [27]:
```
#using existing dataframe and formate it so no duplicates remain and every olympic game appears once
games=sports_events[['Games', 'Year', 'Season', 'City']].drop_duplicates()
```

In [28]:
```
#sorting values based on Games
games.sort_values('Games', inplace=True)
```

In [29]:
```
#using lambda to allocate to summer and winter games an unique number/ID
games['Season_ID']=games['Season'].apply(lambda x: 1 if x=='Summer' else 2)
```

In [30]:
```
len(games.Games)
```

Out[30]:
```
52
```

In [31]:
```
#allocating every game an unique number/ID
games['Games_ID']=range(1,53)
```

In [32]:
```
#droping column 'Season', this could have also been done with reindex()
games.drop('Season',1, inplace=True)
```

```
/var/folders/06/r72gs4510ldd1g4t75zm78vc0000gn/T/ipykernel_4326/2062314925.py:1: FutureWarning: In a future ver
sion of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only
  games.drop('Season',1, inplace=True)
```

In [33]:
```
games.head()
```

Out[33]:

	Games	Year	City	Season_ID	Games_ID
3079	1896 Summer	1896	Athina	1	1
3	1900 Summer	1900	Paris	1	2
711	1904 Summer	1904	St. Louis	1	3
268	1906 Summer	1906	Athina	1	4
1149	1908 Summer	1908	London	1	5

In [34]:
```
#reindexing the columns
games_df=games.reindex(columns=['Games', 'Games_ID', 'Year', 'Season_ID', 'City'])
```

In [35]:
```
#setting a new index
games_df.set_index('Games_ID', inplace=True)
```

In [36]:
```
#saving it on a csv file
games_df.to_csv('/Users/alexandergandji/Desktop/Course Folder/Capstone Project SQL Data Science /Sports/olympic
```

athletes

In [37]:
```
sports_events.head()
```

Out[37]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN

In [38]:
```
#getting an unique list of athletes
athlete=sports_events.Name.unique().tolist()
```

In [39]:
```
#creating a new dataframe
athlete=pd.DataFrame(athlete, columns=['Name'])
```

In [40]:
```
#sorting dataframe by name in ascending order
athlete.sort_values('Name', inplace=True)
```

In [42]:
```
#removing the extra space
athlete.replace('  Gabrielle Marie "Gabby" Adcock (White-)', 'Gabrielle Marie "Gabby" Adcock (White-)',
                inplace=True)
athlete.replace(' Eleonora Margarida Josephina Scmitt', 'Eleonora Margarida Josephina Scmitt', inplace=True)
athlete.replace(' Jean Hauptmanns', 'Jean Hauptmanns', inplace=True)
athlete.replace(' Luis ngel Fernando de los Santos Grossi', 'Luis ngel Fernando de los Santos Grossi', inplace=
athlete.replace(' Th Anh', 'Th Anh', inplace=True)
athlete.replace(' Th Ngn Thng', 'Th Ngn Thng', inplace=True)
athlete.replace(' Tin Tun', 'Tin Tun', inplace=True)
athlete.sort_values('Name', inplace=True)
```

In [105…
```
#allocating an unique number/ID to every athlete and merging the dataframe with the orginal data to get the gen
athlete['Athlete_ID']=range(1,134788)
athlete=athlete.merge(sports_events[['Name', 'Sex']], how='left').drop_duplicates()
```

In [106…
```
#using lambda to allocate an unique number to every gender
athlete['Sex_ID']=athlete.Sex.apply(lambda x: 1 if x=='M' else 2)
```

In [108…
```
#setting a new index
athlete.set_index('Athlete_ID', inplace=True)
```

In [151…
```
ath1=athlete.dropna()
```

In [153…
```
#saving it on a csv file
ath1.to_csv('/Users/alexandergandji/Desktop/Course Folder/Capstone Project SQL Data Science /Sports/athlete.csv
```

Sports

In [48]:
```
sports_events.head()
```

Out[48]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
0	1	A Dijiang	M	24.0	180.0	80.0	China	CHN	1992 Summer	1992	Summer	Barcelona	Basketball	Basketball Men's Basketball	NaN
1	2	A Lamusi	M	23.0	170.0	60.0	China	CHN	2012 Summer	2012	Summer	London	Judo	Judo Men's Extra-Lightweight	NaN
2	3	Gunnar Nielsen Aaby	M	24.0	NaN	NaN	Denmark	DEN	1920 Summer	1920	Summer	Antwerpen	Football	Football Men's Football	NaN
3	4	Edgar Lindenau Aabye	M	34.0	NaN	NaN	Denmark/Sweden	DEN	1900 Summer	1900	Summer	Paris	Tug-Of-War	Tug-Of-War Men's Tug-Of-War	Gold
4	5	Christine Jacoba Aaftink	F	21.0	185.0	82.0	Netherlands	NED	1988 Winter	1988	Winter	Calgary	Speed Skating	Speed Skating Women's 500 metres	NaN

In [49]:
```
#creating a new dataframe for every sport event
sports=pd.DataFrame(sports_events.Event.unique().tolist(), columns=['Sport'])
```

In [50]:
```
#sorting data by sport events
sports.sort_values('Sport', inplace=True)
```

In [51]:
```
sports
```

Out[51]:

	Sport
764	Aeronautics Mixed Aeronautics
34	Alpine Skiing Men's Combined
30	Alpine Skiing Men's Downhill
32	Alpine Skiing Men's Giant Slalom
33	Alpine Skiing Men's Slalom
...	...
596	Wrestling Women's Flyweight, Freestyle
340	Wrestling Women's Heavyweight, Freestyle
292	Wrestling Women's Light-Heavyweight, Freestyle
356	Wrestling Women's Lightweight, Freestyle
609	Wrestling Women's Middleweight, Freestyle

765 rows × 1 columns

In [52]:
```
#allocating an unique ID to every sport event
sport=sports['Sport_ID']=range(1,766)
```

In [53]:
```
#setting a new index
sports.set_index('Sport_ID', inplace=True)
```

In [54]:
```
sports
```

Out[54]:

	Sport
Sport_ID	
1	Aeronautics Mixed Aeronautics
2	Alpine Skiing Men's Combined
3	Alpine Skiing Men's Downhill
4	Alpine Skiing Men's Giant Slalom
5	Alpine Skiing Men's Slalom
...	...
761	Wrestling Women's Flyweight, Freestyle
762	Wrestling Women's Heavyweight, Freestyle
763	Wrestling Women's Light-Heavyweight, Freestyle
764	Wrestling Women's Lightweight, Freestyle
765	Wrestling Women's Middleweight, Freestyle

765 rows × 1 columns

In [55]:
```
#saving it on a csv file
sports.to_csv('/Users/alexandergandji/Desktop/Course Folder/Capstone Project SQL Data Science /Sports/sports.cs
```

results

In [248…
```
#merging orginal data with the nations dataframe
results1=sports_events.merge(nat_df1.reset_index(), how='left')
```

In [249…
```
#adding a new column, which contains the medal id; using lambda
results1['Medal_ID']=results1.Medal.apply(lambda x: 1 if x=='Gold' else
                                          (2 if x=='Silver' else (3 if x=='Bronze' else 4)))
```

In [250…
```
#removing extra space from athletes name
results1.replace('  Gabrielle Marie "Gabby" Adcock (White-)', 'Gabrielle Marie "Gabby" Adcock (White-)',
                 inplace=True)
results1.replace(' Eleonora Margarida Josephina Scmitt', 'Eleonora Margarida Josephina Scmitt', inplace=True)
results1.replace(' Jean Hauptmanns', 'Jean Hauptmanns', inplace=True)
results1.replace(' Luis ngel Fernando de los Santos Grossi', 'Luis ngel Fernando de los Santos Grossi',
                 inplace=True)
results1.replace(' Th Anh', 'Th Anh', inplace=True)
results1.replace(' Th Ngn Thng', 'Th Ngn Thng', inplace=True)
results1.replace(' Tin Tun', 'Tin Tun', inplace=True)
```

In [251…
```
#merging results dataframe with the games dataframe
results2=results1.merge(games_df.reset_index(), how='left')
```

In [252…
```
#merging results dataframe with the athlete dataframe
results3=results2.merge(athlete.reset_index(), how='left')
```

In [253…
```
#dropping 'Sport' column=> this column just gave an general overview of the events
results3.drop('Sport',1,inplace=True)
```

```
/var/folders/06/r72gs4510ldd1g4t75zm78vc0000gn/T/ipykernel_4326/93642161.py:1: FutureWarning: In a future versi
on of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only
  results3.drop('Sport',1,inplace=True)
```

In [254…
```
#renaming Event to Sport=> the new Sport column has a more detailed overview of the different events
results3.rename(columns={'Event':'Sport'}, inplace=True)
```

In [255…
```
#merging results dataframe with sports dataframe
results4=results3.merge(sports.reset_index(), how='left')
```

In [256…
```
#sort result dataframe by Year
results4.sort_values('Year', inplace=True)
```

In [257…
```
results4
```

Out[257]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	...	City	Sport	Medal	NOC
68791	35094	Angelos Fetsis	M	NaN	NaN	NaN	Greece	GRE	1896 Summer	1896	...	Athina	Athletics Men's 800 metres	NaN	
204735	102806	Richard Rstel	M	NaN	NaN	NaN	Germany	GER	1896 Summer	1896	...	Athina	Gymnastics Men's Horizontal Bar, Teams	Gold	
181753	91353	Georgios Paraskevopoulos	M	NaN	NaN	NaN	Greece	GRE	1896 Summer	1896	...	Athina	Cycling Men's Road Race, Individual	NaN	
181754	91353	Georgios Paraskevopoulos	M	NaN	NaN	NaN	Greece	GRE	1896 Summer	1896	...	Athina	Cycling Men's 12-Hours Race	NaN	
204736	102806	Richard Rstel	M	NaN	NaN	NaN	Germany	GER	1896 Summer	1896	...	Athina	Gymnastics Men's Pommelled Horse	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
40051	20599	Phillip Hung Chew	M	22.0	173.0	91.0	United States	USA	2016 Summer	2016	...	Rio de Janeiro	Badminton Men's Doubles	NaN	
194190	97481	Bence Pulai	M	24.0	193.0	89.0	Hungary	HUN	2016 Summer	2016	...	Rio de Janeiro	Swimming Men's 100 metres Butterfly	NaN	
194191	97481	Bence Pulai	M	24.0	193.0	89.0	Hungary	HUN	2016 Summer	2016	...	Rio de Janeiro	Swimming Men's 4 x 100 metres Medley Relay	NaN	
40081	20616	Danny Chia	M	43.0	170.0	75.0	Malaysia	MAS	2016 Summer	2016	...	Rio de Janeiro	Golf Men's Individual	NaN	
245289	122828	Nigora Tursunkulova	F	17.0	181.0	67.0	Uzbekistan	UZB	2016 Summer	2016	...	Rio de Janeiro	Taekwondo Women's Welterweight	NaN	

271119 rows × 21 columns

In [258…
```
#due to the fact that we have data points which did not have any value in the Age, Height and Weight column we
#used ffill()
results4['Age']=results4.Age.ffill(axis=0)
#applying ffill() in the first row of the Age column
results4.Age.replace(results4.Age.iloc[0],21, inplace=True)
#transforming Age from float to integer
results4['Age']=results4.Age.apply(np.int64)
```

In [259…
```
results4=results4.copy()
```

In [260…
```
#applying ffill() in the first row of the Height column
results4['Height']=results4.Height.replace(results4.Height.iloc[0],176)
#transforming Height from float to integer
results4['Height']=results4.Height.apply(np.int64)

### 176 was chosen, because the height was found at other athletes in the same time period and disceplines
```

In [261…
```
results4
```

Out[261]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	...	City	Sport	Medal	NOC
68791	35094	Angelos Fetsis	M	21	176	NaN	Greece	GRE	1896 Summer	1896	...	Athina	Athletics Men's 800 metres	NaN	
204735	102806	Richard Rstel	M	21	176	NaN	Germany	GER	1896 Summer	1896	...	Athina	Gymnastics Men's Horizontal Bar, Teams	Gold	
181753	91353	Georgios Paraskevopoulos	M	21	176	NaN	Greece	GRE	1896 Summer	1896	...	Athina	Cycling Men's Road Race, Individual	NaN	
181754	91353	Georgios Paraskevopoulos	M	21	176	NaN	Greece	GRE	1896 Summer	1896	...	Athina	Cycling Men's 12-Hours Race	NaN	
204736	102806	Richard Rstel	M	21	176	NaN	Germany	GER	1896 Summer	1896	...	Athina	Gymnastics Men's Pommelled Horse	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
40051	20599	Phillip Hung Chew	M	22	173	91.0	United States	USA	2016 Summer	2016	...	Rio de Janeiro	Badminton Men's Doubles	NaN	
194190	97481	Bence Pulai	M	24	193	89.0	Hungary	HUN	2016 Summer	2016	...	Rio de Janeiro	Swimming Men's 100 metres Butterfly	NaN	
194191	97481	Bence Pulai	M	24	193	89.0	Hungary	HUN	2016 Summer	2016	...	Rio de Janeiro	Swimming Men's 4 x 100 metres Medley Relay	NaN	
40081	20616	Danny Chia	M	43	170	75.0	Malaysia	MAS	2016 Summer	2016	...	Rio de Janeiro	Golf Men's Individual	NaN	
245289	122828	Nigora Tursunkulova	F	17	181	67.0	Uzbekistan	UZB	2016 Summer	2016	...	Rio de Janeiro	Taekwondo Women's Welterweight	NaN	

271119 rows × 21 columns

In [262…
```
x=round((66+66+69)/3,0)
print(x)
### These 3 values are representive for the weight in the early time period, early 20th century, and disceplien
```

```
67.0
```

In [263…
```
#applying ffill() in the first row of the Weight column
results4['Weight']=results4.Weight.replace(results4.Weight.iloc[0],67)
#transforming Weight from float to integer
results4['Weight']=results4.Weight.apply(np.int64)
```

In [264…
```
results4[results4.Athlete_ID.isnull()]
```

Out[264]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	...	City	Sport	Medal	NOC_ID
92100	46661	Jean Hauptmanns	M	26	176	67	Germany	GER	1912 Summer	1912	...	Stockholm	Wrestling Men's Heavyweight, Greco-Roman	NaN	80
51334	26388	Luis ngel Fernando de los Santos Grossi	M	23	176	67	Uruguay	URU	1948 Summer	1948	...	London	Cycling Men's Team Pursuit, 4,000 metres	NaN	216
215056	107953	Eleonora Margarida Josephina Scmitt	F	16	176	67	Brazil	BRA	1948 Summer	1948	...	London	Swimming Women's 4 x 100 metres Freestyle Relay	NaN	30
215055	107953	Eleonora Margarida Josephina Scmitt	F	16	176	67	Brazil	BRA	1948 Summer	1948	...	London	Swimming Women's 100 metres Freestyle	NaN	30
51335	26388	Luis ngel Fernando de los Santos Grossi	M	27	176	67	Uruguay	URU	1952 Summer	1952	...	Helsinki	Cycling Men's Road Race, Individual	NaN	216
51336	26388	Luis ngel Fernando de los Santos Grossi	M	27	176	67	Uruguay	URU	1952 Summer	1952	...	Helsinki	Cycling Men's Road Race, Team	NaN	216
51337	26388	Luis ngel Fernando de los Santos Grossi	M	27	176	67	Uruguay	URU	1952 Summer	1952	...	Helsinki	Cycling Men's 1,000 metres Time Trial	NaN	216
51338	26388	Luis ngel Fernando de los Santos Grossi	M	27	176	67	Uruguay	URU	1952 Summer	1952	...	Helsinki	Cycling Men's Team Pursuit, 4,000 metres	NaN	216
58302	29843	Th Ngn Thng	F	19	147	47	Vietnam	VIE	2008 Summer	2008	...	Beijing	Gymnastics Women's Individual All-Around	NaN	22
58303	29843	Th Ngn Thng	F	19	147	47	Vietnam	VIE	2008 Summer	2008	...	Beijing	Gymnastics Women's Floor Exercise	NaN	22
58304	29843	Th Ngn Thng	F	19	147	47	Vietnam	VIE	2008 Summer	2008	...	Beijing	Gymnastics Women's Horse Vault	NaN	22
58305	29843	Th Ngn Thng	F	19	147	47	Vietnam	VIE	2008 Summer	2008	...	Beijing	Gymnastics Women's Uneven Bars	NaN	22
58306	29843	Th Ngn Thng	F	19	147	47	Vietnam	VIE	2008 Summer	2008	...	Beijing	Gymnastics Women's Balance Beam	NaN	22
58308	29843	Th Ngn Thng	F	23	147	47	Vietnam	VIE	2012 Summer	2012	...	London	Gymnastics Women's Balance Beam	NaN	22
58307	29843	Th Ngn Thng	F	23	147	47	Vietnam	VIE	2012 Summer	2012	...	London	Gymnastics Women's Uneven Bars	NaN	22
1560	869	Gabrielle Marie "Gabby" Adcock (White-)	F	25	167	67	Great Britain	GBR	2016 Summer	2016	...	Rio de Janeiro	Badminton Mixed Doubles	NaN	75
58301	29842	Th Anh	F	20	165	58	Vietnam	VIE	2016 Summer	2016	...	Rio de Janeiro	Fencing Women's Foil, Individual	NaN	22

17 rows × 21 columns

In [265…
```
#these data points were dropped, because even after the use of ffill(), the data points were too ambigous
results4.drop([92100, 58301, 1560, 58307, 58308, 51338, 58306, 58305, 58304, 58303, 58302,
               51337, 51336, 51335, 51334, 215055, 215056], 0, inplace=True)
```

```
/var/folders/06/r72gs4510ldd1g4t75zm78vc0000gn/T/ipykernel_4326/564590031.py:1: FutureWarning: In a future vers
ion of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only
  results4.drop([92100, 58301, 1560, 58307, 58308, 51338, 58306, 58305, 58304, 58303, 58302, 51337, 51336, 5133
5, 51334, 215055, 215056], 0, inplace=True)
```

In [266…
```
results4
```

Out[266]:

	ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	...	City	Sport	Medal	NOC
68791	35094	Angelos Fetsis	M	21	176	67	Greece	GRE	1896 Summer	1896	...	Athina	Athletics Men's 800 metres	NaN	
204735	102806	Richard Rstel	M	21	176	67	Germany	GER	1896 Summer	1896	...	Athina	Gymnastics Men's Horizontal Bar, Teams	Gold	
181753	91353	Georgios Paraskevopoulos	M	21	176	67	Greece	GRE	1896 Summer	1896	...	Athina	Cycling Men's Road Race, Individual	NaN	
181754	91353	Georgios Paraskevopoulos	M	21	176	67	Greece	GRE	1896 Summer	1896	...	Athina	Cycling Men's 12-Hours Race	NaN	
204736	102806	Richard Rstel	M	21	176	67	Germany	GER	1896 Summer	1896	...	Athina	Gymnastics Men's Pommelled Horse	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
40051	20599	Phillip Hung Chew	M	22	173	91	United States	USA	2016 Summer	2016	...	Rio de Janeiro	Badminton Men's Doubles	NaN	
194190	97481	Bence Pulai	M	24	193	89	Hungary	HUN	2016 Summer	2016	...	Rio de Janeiro	Swimming Men's 100 metres Butterfly	NaN	
194191	97481	Bence Pulai	M	24	193	89	Hungary	HUN	2016 Summer	2016	...	Rio de Janeiro	Swimming Men's 4 x 100 metres Medley Relay	NaN	
40081	20616	Danny Chia	M	43	170	75	Malaysia	MAS	2016 Summer	2016	...	Rio de Janeiro	Golf Men's Individual	NaN	
245289	122828	Nigora Tursunkulova	F	17	181	67	Uzbekistan	UZB	2016 Summer	2016	...	Rio de Janeiro	Taekwondo Women's Welterweight	NaN	

271102 rows × 21 columns

In [267…
```
#transforming Athlete_ID from float to integer
results4['Athlete_ID']=results4.Athlete_ID.apply(np.int64)
```

In [268…
```
#reindexing the columns
results5=results4.reindex(columns=['Athlete_ID', 'Sex', 'Age', 'Height', 'Weight', 'NOC_ID',
                                   'Games_ID', 'Year', 'Season_ID', 'City', 'Sport_ID', 'Medal_ID'])
```

In [269…
```
#setting a new index
results5.set_index('Athlete_ID', inplace=True)
```

In [270…
```
results5
```

Out[270]:

	Sex	Age	Height	Weight	NOC_ID	Games_ID	Year	Season_ID	City	Sport_ID	Medal_ID
Athlete_ID											
8910	M	21	176	67	82	1	1896	1	Athina	102	4
105935	M	21	176	67	80	1	1896	1	Athina	367	1
40872	M	21	176	67	82	1	1896	1	Athina	278	4
40872	M	21	176	67	82	1	1896	1	Athina	261	4
105935	M	21	176	67	80	1	1896	1	Athina	376	4
...	...	...	...	...	...	...	...	...	...	...	...
101394	M	22	173	91	217	52	2016	1	Rio de Janeiro	154	4
14303	M	24	193	89	91	52	2016	1	Rio de Janeiro	632	4
14303	M	24	193	89	91	52	2016	1	Rio de Janeiro	645	4
24827	M	43	170	75	127	52	2016	1	Rio de Janeiro	361	4
93320	F	17	181	67	218	52	2016	1	Rio de Janeiro	696	4

271102 rows × 11 columns

In [271…
```
#adding a new column, with unique values for every gender
results5['Sex_ID']=results5['Sex'].apply(lambda x: 1 if x=='M' else 2)
```

In [272…
```
#dropping the 'Sex' column
results5.drop('Sex',1, inplace=True)
```

```
/var/folders/06/r72gs4510ldd1g4t75zm78vc0000gn/T/ipykernel_4326/426421636.py:1: FutureWarning: In a future vers
ion of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only
  results5.drop('Sex',1, inplace=True)
```

In [273…
```
#saving it on a csv file
results5.to_csv('/Users/alexandergandji/Desktop/Course Folder/Capstone Project SQL Data Science /Sports/results
```