**FINAL PROJECT - Kelley, Anna, Kyle, Justin, Kelvin, Paige**

**#1 - What range of years does the provided database cover?**

```
create temporary table Combined AS
select yearID from allstarfull
union
select yearID from appearances
union
select yearID from awardsmanagers
union
select yearID from awardsplayers
union
select yearID from awardssharemanagers
union
select yearID from awardsshareplayers
union
select yearID from batting
union
select yearID from battingpost
union
select yearID from collegeplaying
union
select yearID from fielding
union
select yearID from fieldingof
union
select yearID from fieldingofsplit
union
select yearID from fieldingpost
union
select yearID from halloffame
union
select yearID from managers
union
select yearID from managershalf
union
select yearID from pitching
union
select yearID from pitchingpost
union
select yearID from salaries
union
select yearID from seriespost
```

```
union
select yearID from teams
union
select yearID from teamshalf;


select MIN(yearID), MAX(yearID) from Combined;
```

**Answer:** 1864 to 2019

**#2 - Find the name and height of the shortest player in the database. How many games did he play in? What is the name of the team for which he played?**

```
SELECT nameFirst, nameLast, teamID, height, G_all
FROM people
INNER JOIN appearances
ON people.playerID = appearances.playerID
WHERE height IS NOT NULL
ORDER BY height;
SELECT name
FROM teams
WHERE teamID = 'SLA';
```

**Answer:** Eddie Gaedel, height = 43, G_all = 1

**#3** - **Find all players in the database who played at Vanderbilt University. Create a list showing each player's first and last names as well as the total salary they earned in the major leagues.**
**#Sort this list in descending order by the total salary earned. Which Vanderbilt player earned the most money in the majors?**

```
SELECT schoolID
FROM schools
WHERE name_full = 'Vanderbilt University';
SELECT people.nameFirst, people.nameLast, collegeplaying.schoolID, SUM(salary)
FROM people
INNER JOIN salaries ON salaries.playerID = people.playerID
INNER JOIN collegeplaying ON collegeplaying.playerID = people.playerID
WHERE collegeplaying.schoolID = 'vandy'
GROUP BY people.nameFirst,people.nameLast,schoolID
ORDER BY SUM(salary)desc;
```

**Answer:** David Price earned the most at 245,553,888

**#5 - Find the average number of strikeouts per game by decade since 1920. Round the numbers you report to 2 decimal places. Do the same for home runs per game. Do you see any trends?**

```
SELECT CONCAT(SUBSTRING(yearID, 1, 3), 0)  AS decade,
        ROUND(AVG(SO),2) AS strikeout,
   ROUND(AVG(HR),2) AS homeruns
FROM pitching
WHERE (yearid*10)/10 >= 1920
GROUP BY decade;
```

**Answer:** Baseball was desegregated in 1947, appearing to lead to a higher number of strikeouts and home runs from 1960 onward.

**#6 - Find the player who had the most success stealing bases in 2016, where success is measured as the percentage of stolen base attempts which are successful.**
**(A stolen base attempt results either in a stolen base or being caught stealing.) Consider only players who attempted at least 20 stolen bases.**

```
SELECT nameFirst,nameLast, SUCCESS
FROM(SELECT playerID, yearID, SB, ATTEMPTS, ROUND(SB/ATTEMPTS* 100,2) AS
SUCCESS
FROM(SELECT SB, yearID, playerID, SUM(SB + CS) AS ATTEMPTS
FROM batting
WHERE yearID =2016
GROUP BY SB, yearID, playerID) STOLEN
WHERE ATTEMPTS >= 20
ORDER BY SUCCESS DESC) STOLEN2
LEFT JOIN people
ON STOLEN2.playerID =people.playerID
ORDER BY SUCCESS DESC;
```

**Answer:** Chris Owings has the most success at 91%

**#8** - **Using the attendance figures from the homegames table, find the teams and parks which had the top 5 average attendance per game in 2016 (where average attendance is defined as total attendance divided by number of games).**
**##Only consider parks where there were at least 10 games played. Report the park name, team name, and average attendance. Repeat for the lowest 5 average attendance.**

```
SELECT t.name, p.parkname, h.attendance, h.games,
ROUND(CAST(h.attendance as dec) / CAST(h.games as dec), 1) as avg_attendance
FROM homegames h
```

```
LEFT JOIN parks p ON h.parkkey = p.parkkey
LEFT JOIN teams as t ON t.yearid = h.yearkey AND h.teamkey = t.teamid
WHERE yearid = 2016 AND h.games >= 10
ORDER BY avg_Attendance desc;
```

**Answer:** Dodgers, Cardinals, Blue Jays, Giants, and Cubs

**Group 1**: **Is there any correlation between number of wins and team salary? Use data from 2000 and later to answer this question. As you do this analysis, keep in mind that salaries across the whole league tend to increase together, so you may want to look on a year-by-year basis.**

```
SELECT s.yearid, s.teamid, t.name, team_salary, t.w as wins
FROM (SELECT salaries.yearid, salaries.teamid, SUM(salary) as team_salary
FROM salaries
WHERE salaries.yearid >= 2000
GROUP BY salaries.yearid, salaries.teamid) s
LEFT JOIN teams t on s.teamid = t.teamid AND s.yearid = t.yearid
ORDER BY name;
```

**Answer: After pulling this report and exporting to Python to visualize, I cannot tell that there is a strong trend in salary to wins. There are teams that are paid significantly more but they seem to have a relatively even chance of winning compared to other teams. The main thing we can determine from this is that the salary of the team is based on other factors. We did note that a series of high win years in a row could create a boost in pay but that did not necessarily result in continued high win rates.**

In [3]:

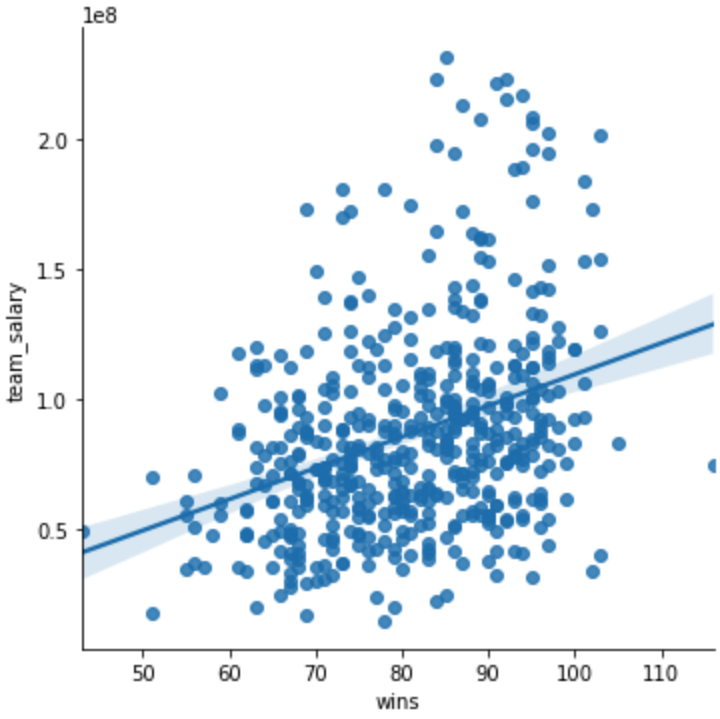
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

team_salary= pd.read_csv('team_salary')
team_salary.head()
```

Out[3]:

| | yearid | teamid | name | team_salary | wins |
|---|---|---|---|---|---|
| **0** | 2000 | ANA | Anaheim Angels | 51464167 | 82 |
| **1** | 2002 | ANA | Anaheim Angels | 61721667 | 99 |
| **2** | 2004 | ANA | Anaheim Angels | 100534667 | 92 |
| **3** | 2001 | ANA | Anaheim Angels | 47535167 | 75 |
| **4** | 2003 | ANA | Anaheim Angels | 79031667 | 77 |

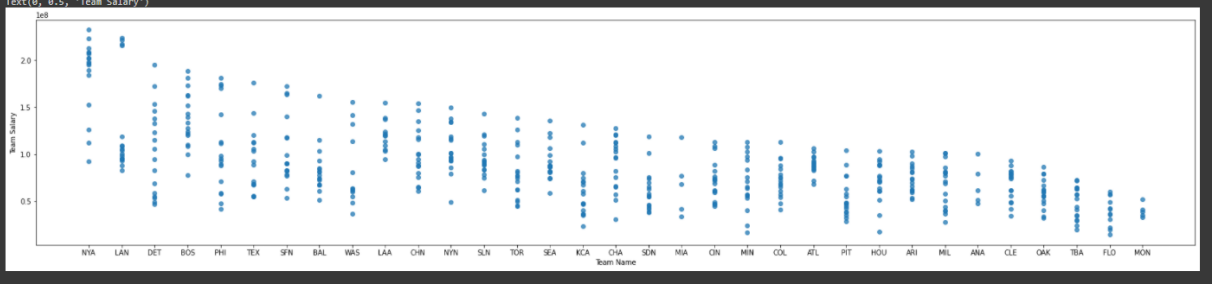
In [4]:

```
sns.lmplot(x="wins", y="team_salary", data=team_salary)
plt.show()
```



```
fig = plot.subplots(figsize = (30,6))

plot.scatter(x = df['teamid'],
         y = df['team_salary'],
           alpha = 0.75, )

plot.xlabel('Team Name')
plot.ylabel('Team Salary')
```

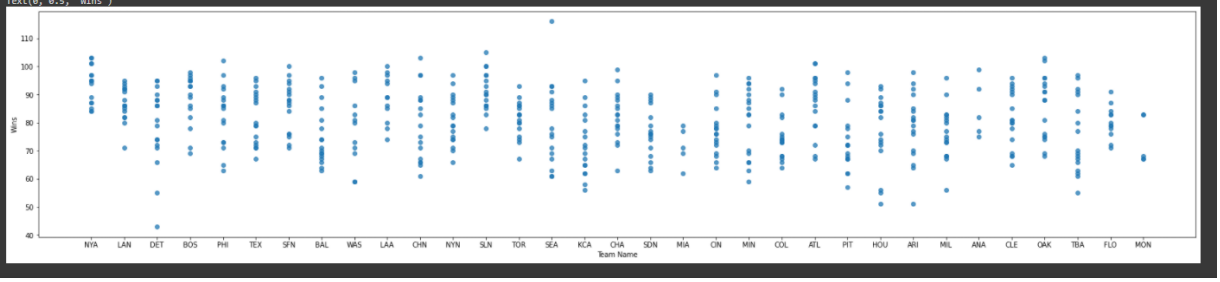
```
Text(0, 0.5, 'Team Salary')
```




## Teams to Salary

```
fig = plot.subplots(figsize = (30,6))

plot.scatter(x = df['teamid'],
         y = df['wins'],
           alpha = 0.75, )

plot.xlabel('Team Name')
plot.ylabel('Wins')
```

```
Text(0, 0.5, 'Wins')
```




## Teams to Wins

```
SELECT s.yearid, s.teamid, t.name, team_salary, t.w as wins
FROM (SELECT salaries.yearid, salaries.teamid, SUM(salary) as team_salary
FROM salaries
WHERE salaries.yearid >= 2000
GROUP BY salaries.yearid, salaries.teamid) s
LEFT JOIN teams t on s.teamid = t.teamid AND s.yearid = t.yearid
ORDER BY name;
```