Introduction- I wanted to continue to practice my SQL skills and I thought it would be fun to create an all-time baseball lineup where each player’s first name was some iteration of my first name, Will, based solely on their career home run numbers. To do so, I decided to download the History of Baseball dataset on Kaggle and upload the files to SQL for analysis.

Datasets – After searching through the multiple csv files found within the dataset, I decided that I should use the players.csv, appearances.csv, and batting.csv files.

The columns needed for this analysis from each dataset were as follows:

Players.csv – player\_id, name\_first, name\_last

Appearances.csv – player\_id, g\_c, g\_1b, g\_2b, g\_3b, g\_ss, g\_lf, g\_cf, g\_rf

Batting.csv – player\_id, hr

Dataset Limitations - Unfortunately, this dataset only had player statistics through the 2015 season for each player. This ended up leading to players who would have made the team being left off the roster. They were included in the honorable mentions section of this analysis but are not included on the roster because they were not included in the dataset.

Setup – I needed to consolidate all the data available into one table that would include the player\_id, name\_first, name\_last, career\_hr and total games played at each position to determine the position they would play on the team.

-- All players with first name Will. This would include all players with an version of my name (Will, Wil, William, Willie, Wilson, etc.) The query used was:

SELECT player\_id, name\_first, name\_last

FROM `baseball-will-team.baseball.players`

WHERE name\_first LIKE '%Wil%';

-- The results brought back 160 players. This query was then saved as its own table “wills” for future analysis.

-- I needed to combine the existing player data with the batting data so I could see the player’s full name along with their batting statistics. The query used was:

SELECT \*

FROM `baseball-will-team.baseball.batting` bat

JOIN `baseball-will-team.baseball.players` players

ON bat.player\_id = players.player\_id;

This query resulted in a new table “stats” that would be used in later queries.

-- Next, I needed every player’s career home run totals. The dataset, “stats”, listed each players yearly statistics and I needed to combine each season’s homerun totals into a career total. The query used was:

SELECT player\_id, SUM(hr) AS career\_hr

FROM `baseball-will-team.baseball.stats`

GROUP BY player\_id

ORDER BY SUM(hr) DESC;

The result was every player’s career homerun total in decending order. This query was then saved as its own table, “hr”.

-- Will’s Career HR totals

SELECT \*

FROM `baseball-will-team.baseball.wills` wills

JOIN `baseball-will-team.baseball.hr` hr

ON wills.player\_id = hr.player\_id

ORDER BY career\_hr DESC;

This query resulted in creating a table, “wills\_career\_hr”, and included 156 players. 4 players were left off because they had no homeruns in their career.

-- Now that I had the career homerun totals for each Will, I needed to form a starting lineup. This meant I needed to know each player’s games played at each possible position. The query used to answer this question was:

SELECT player\_id, SUM(g\_1b) as g\_1B, SUM(g\_2b) as g\_2B, SUM(g\_3b) as g\_3B, SUM(g\_ss) as g\_SS, SUM(g\_lf) AS g\_LF, SUM(g\_cf) as g\_CF, SUM(g\_rf) as g\_RF, SUM(g\_c) as g\_C, SUM(g\_all) as g\_total

FROM `baseball-will-team.baseball.appearances`

GROUP BY player\_id

ORDER BY player\_id DESC;

The query resulted in a table listing each player’s total games played at each position. The table was then saved as “positions” to be used for later analysis.

-- Almost there. It was time to combine all the tables I had created with the data I needed in order to make my final analysis. To do so, I needed to combine the career\_hr table and the positions table. The query used was:

SELECT \*

FROM `baseball-will-team.baseball.wills\_career\_hr` wills

JOIN `baseball-will-team.baseball.positions` positions

ON wills.player\_id = positions.player\_id

This query resulted in a table that included the player\_id, name\_first, name\_last, career\_hr and the total of every game played at each position. The table was saved as “FINAL\_TABLE” to be used for the final analysis.

-- Analysis

Time to build a team. The criteria was first the number of homeruns a player hit over their career, followed by the number of games played at a specific position. Each player needed to have played atleast 100 games at a position to be qualified for that position on the roster. The query used was:

SELECT \*

FROM `baseball-will-team.baseball.FINAL\_TABLE`

WHERE g\_C > 100

ORDER BY g\_C, career\_hr DESC;

This query was repeated for each position. The WHERE and first ORDER BY clause, example “g\_c”, was changed for each query to find the player who played each position (g\_1B = 1B, g\_2B = 2B, g\_3B = 3B, etc.)

-- Final Squad.

Players in () are honorable mentions who would have technically made the team, but their careers either started after 2015 or they did not accumulate enough stats by the end of 2015 to count in this analysis.

C – Wilson Ramos HR = 61,

(C – Will Smith HR = 84)

(C – Wilson Contreras HR = 125)

1B – Willie McCovey HR = 521

2B – Willie Randolph HR = 54

3B – Willie Jones HR = 190

SS – Wil Cordero HR = 122

LF – Willie Stargell HR = 475

CF – Willie Mays HR = 660

RF – Willie Kirkland HR = 148

(RF – Wil Myers HR = 156)

DH – Willie Horton HR =325

IF – Will Clark HR = 284

OF – Will Davis HR = 182

Overall, this is a solid lineup. I knew going into this that my starting LF and CF would be Willie Stargell and Willie Mayes, respectively, but I didn’t really know how the rest of the team would shake out. I could add Pitchers to this team in the future, but I just wanted to focus on position players this time around. The biggest take away I received was that if I wanted to be a professional baseball player, I should have gone by Willie, and not Will. At the end of the day, this was a fun, dumb project to complete and I look forward to doing more stuff like this in the future.