Databases Final Project
Phase II
Cover Page (2)

By:
Amritpal Singh: Section 315
Stanley Zheng: Section 415

Emails:
asingh69@jh.edu
szheng22@jh.edu

Application Domain:
The application domain for this project was baseball statistics. We chose to use baseball statistics for this because baseball had many resources to get data from and we were looking to gain insights on one of our favorite sports growing up. We use the Lahman database which is a very comprehensive database covering the course of MLB history since they began taking statistics for players and teams.

How to view our results:
We are submitting a zip file with all of our code in it. We are using a SQLite database for the Lahman Database tables and are running a local server using Python Flask. To be able to run this project and view its contents the user will have to have the ability to use Python 3 on their computer. The requirements to run this project which the user needs to have installed on their computer are flask, sqlite3, pandas, matplotlib, os, sklearn, and numpy libraries, along with Python 3. To run the project go to the root directory of the project which in this case is BaseballDatabase and run the following commands in order:

1. export FLASK_APP=app
2. export FLASK_ENV=development
3. flask run

```
export FLASK_APP=app
export FLASK_ENV=development
flask run
```

At this point you will get a screen that looks like this

```
BaseballDatabase git:(master) ✗ export FLASK_APP=app
➜  BaseballDatabase git:(master) ✗ export FLASK_ENV=development
➜  BaseballDatabase git:(master) ✗ flask run
 * Serving Flask app "app" (lazy loading)
 * Environment: development
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 339-274-723
```

To view the contents of the app copy and paste the http where it says Running on, in this case http://127.0.0.1:5000/, into your browser such as Google Chrome. From there you should be able to access our project.

(3) Changes to Database Design since Phase I:
Our database design stayed essentially the same as we had it in Phase I. The only part which we changed was that we did not have a table for individual games like we had in Phase I. We believed that a table for individual games was too large over the course of MLB history and getting the data would have quite a pain over the course of all of MLB history so instead we focused on teams for individual seasons instead. We also added a table for the all star game statistics for players.

(4) How to load the database with values:
We were able to find a website from which we could download a SQLite database with a comprehensive collection of tables containing results throughout the history of baseball. We talked with Professor Yarowsky about this and he agreed that the database we used, the Lahman database, was a very good resource to get the baseball data from. From the Lahman database we were able to get tables from which we could answer all of our questions and more.
http://www.seanlahman.com/baseball-archive/statistics/

(5) For this project we used a SQLite database. We believed that using a SQLite Database was lighter and provided more flexibility for working together over git and collaboration.

(6) User Guide on how to run the project:
We are submitting a zip file with all of our code in it. We are using a SQLite database for the Lahman Database tables and are running a local server using Python Flask. To be able to run this project and view its contents the user will have to have the ability to use Python 3 on their computer. The requirements to run this project which the user needs to have installed on their computer are flask, sqlite3, pandas, matplotlib, os, sklearn, and numpy libraries, along with Python 3. To run the project go to the root directory of the project which in this case is BaseballDatabase and run the following commands in order:

1. export FLASK_APP=app
2. export FLASK_ENV=development
3. flask run

```
export FLASK_APP=app
export FLASK_ENV=development
flask run
```

At this point you will get a screen that looks like this

```
BaseballDatabase git:(master) ✗ export FLASK_APP=app
➜  BaseballDatabase git:(master) ✗ export FLASK_ENV=development
➜  BaseballDatabase git:(master) ✗ flask run
 * Serving Flask app "app" (lazy loading)
 * Environment: development
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 339-274-723
```

To view the contents of the app copy and paste the http where it says Running on, in this case http://127.0.0.1:5000/, into your browser such as Google Chrome. From there you should be able to access our project.
From there we have a home page where we can go to the homepage by clicking the home value of the navigation bar and the user can go to six different links

1. Stats on Alex Rodriguez
    a. This page provides extensive statistics on famous New York Yankee Alex Rodriguez. It provides statistics in table form and in graph form along with images for every year of his career on his batting, which includes his runs, hits, doubles, triples and home runs vs the average batter in that year with over 450 at bats, along with fielding statistics such as position, games, assists, and errors, all star statistics, awards statistics, and year salary compared to average MLB salary in that year.
2. Does Altitude affect performance study
    a. For this we performed SQL queries to obtain the average batting statistics of the teams at the 6 highest altitude, average batting statistics of teams at the teams at the 6 lowest altitudes, and the average batting statistics for each year since 2000. From there we used pandas and matplotlib to create graphs comparing the batting statistics of the highest altitude, lowest altitude, and average team.
3. The effects of Performance Enhancing Drugs on the game
    a. For this we performed SQL queries which broke up the data into 4 buckets, pre 1974, 1975-1989, 1990-2004, and 2005-present. This was to

compare the batting statistics, the stats which are most recognizable as being affected and helped by PEDS, of the early era of baseball pre 1974, the era before PEDs really took off 1975-1989, the height of the Steroid Scandal 1990-2004, and 2005-present where there are stricter rules against PEDS but the use of them is still very widely spread. We used pandas and matplotlib to create graphs where we compare average runs, hits, doubles, triples, and home runs by each era.

4. Look up team
    a. For this we can look up a team and their statistics in table form by their name or their name and year. Since this web application does not do any error checking the value for the team name must be a valid full name of an MLB team and the year must be a year that the team existed. For example for the Yankees you must put New York Yankees and leave the year input empty and you will get statistics for every year of Yankees history or you put New York Yankees and any year between 1901 to 2019 which are valid years for the Yankees to see single year statistics.
5. Look up player
    a. For this we can look up a player and their statistics in table form along with graphs for their career. Since this web application does not do any error checking the value for the player name must be a valid player that played in the MLB for at least 1 season and the spelling and input style of their name must be valid. For example for Joe DiMaggio the input must be the correct spelling of his full name and it has to be in the form Joe DiMaggio not joe DiMaggio or Joe Dimaggio. A problem that we were having with the look up a player functionality graphs was that the browser would cache the graphs and since for each player the graph name was the same it would show the graphs for a player you searched up previously which were cached. To show that every time you search a player you get different graphs you can look in the static folder for the name of the graph and see that it shares the name of the player you most recently searched up. Another way to see different graphs for different players would be to go into private mode where there are no cached graphs and search up the name of the new player to see that there are different graphs this time.
6. Some cool Data Science
    a. For this I performed some data science techniques which I followed from an online tutorial to create graphs which show some cool insights on the team's data.

(7)
Major Area of Specialization: Data Mining and Knowledge Discovery

For the major area of specialization we focused on Data Mining and Knowledge Discovery. We show this in the graphs we create in the Alex Rodriguez page, the look up a player page, the altitude study page, the PED study page, and the cool data science page. Here we also study the graphs and try to come to an understanding of the data in the graphs. We also create tables that a user can look at to see the exact data from which we created the graphs. Furthermore, in the cool data science page we also go analysis on the distribution of wins throughout MLB history, runs per year on average in MLB history, runs per game vs wins, runs allowed per game vs wins, and performed the k-means machine learning algorithm to gain insights on team wins.

Minor Area of Specialization: Interesting Interface
For the minor area of specialization we created an interesting looking user interface using HTML and CSS Bootstrap which provides a clean and modern web design. We also take in user input and display data according to the user input and provide graphs according to our data.

(8) Give an itemized list of your project's particular strengths or selling points, especially things that may not be obvious upon inspection of your code or output.

1. Provides detailed analysis on games won vs runs and runs allowed
2. Provides the ability to look up player stats and compare them to the league average
3. Provides detailed stats and graphs vs the league average for our favorite player Alex Rodriguez along with salary statistics for him.
4. Provides stats for a team for all years or a single year depending on the input.
5. Does a study on the effects of altitude on batting and provides graphs between locations
6. Does a study on the effects of PEDs on baseball batting and provides graphs between eras

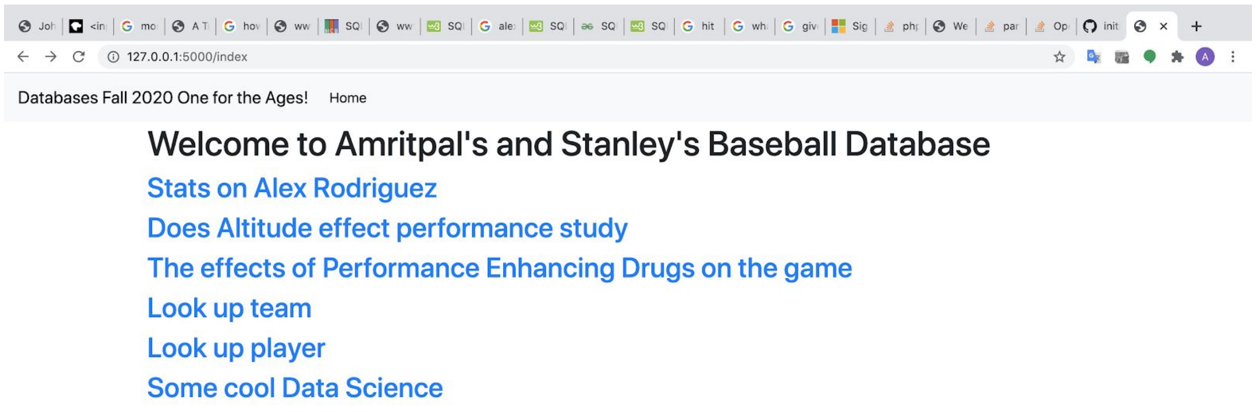
(9) Give a brief itemization of your project's limitations and list suggested possibilities for improvement or worthwhile extension with additional time.

1. The input does not check for bad input and catch errors and thus the program will crash with an error if wrong input is provided in search player and search team. To improve on this we can look into ways in Python or Javascript where we can do input validation and provide a means to let the user know they provided bad input without crashing the program.
2. If given more time we could compare two teams historically and for a single year against each other. This could be used to see how rivalries have historically gone.

3. We could find a way to fix the problem described above where the image is cached in the browser so when the user provides input which updates the image the browser still has the old image cached and looks that up first.

(10)
For the cool data science page I followed a tutorial on data science for baseball statistics and created the graphs myself using data that was compiled from that tutorial.
https://www.datacamp.com/community/tutorials/scikit-learn-tutorial-baseball-1

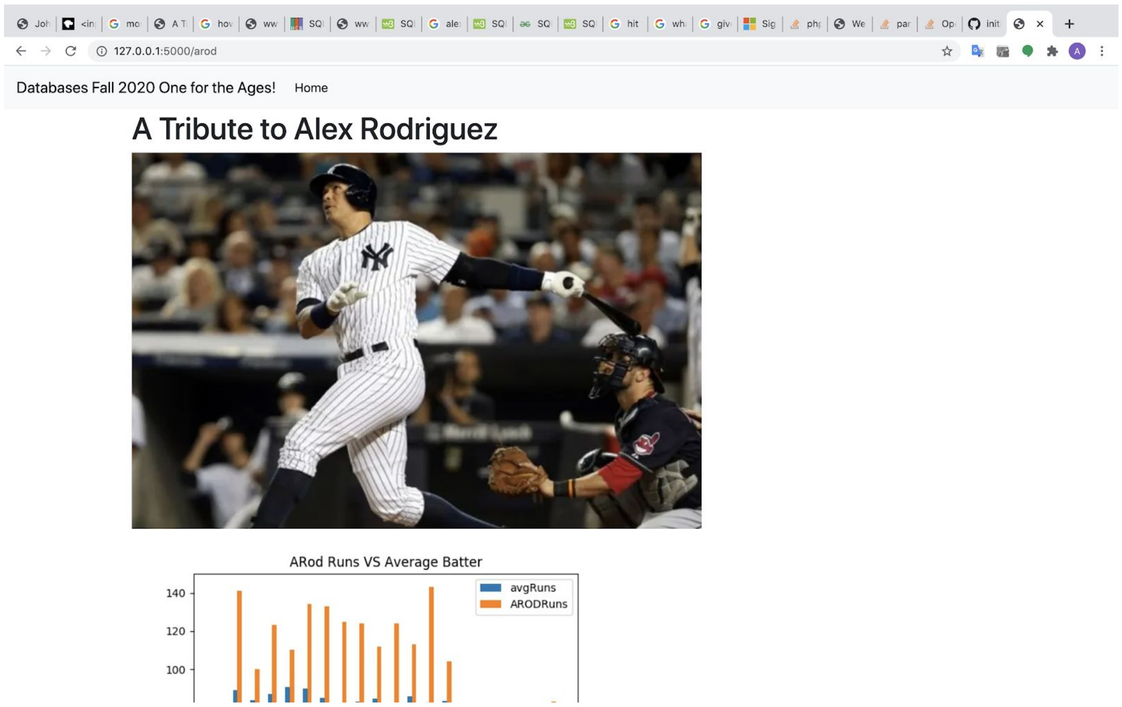
(11)
We start off with our home page from where we can either go home from the navigation bar or select one of six pages to go to.



Welcome to Amritpal's and Stanley's Baseball Database

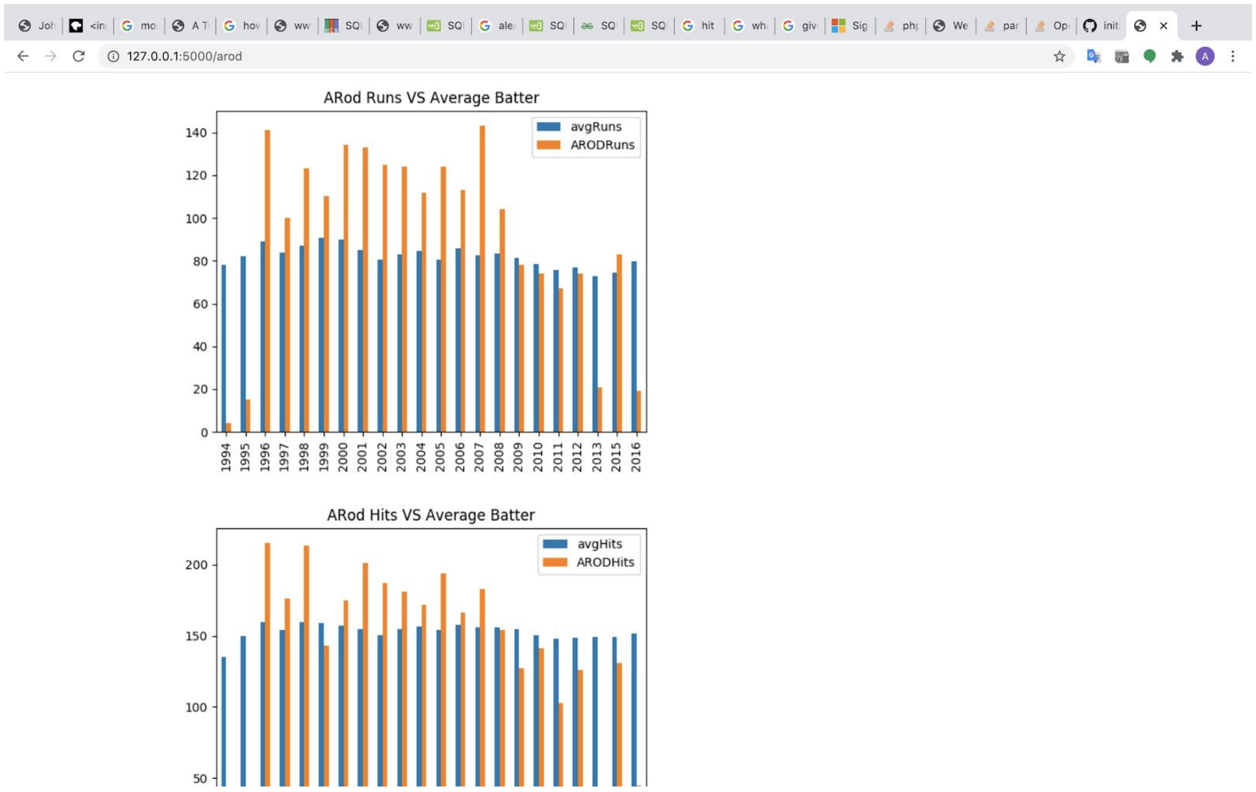
Stats on Alex Rodriguez
Does Altitude effect performance study
The effects of Performance Enhancing Drugs on the game
Look up team
Look up player
Some cool Data Science

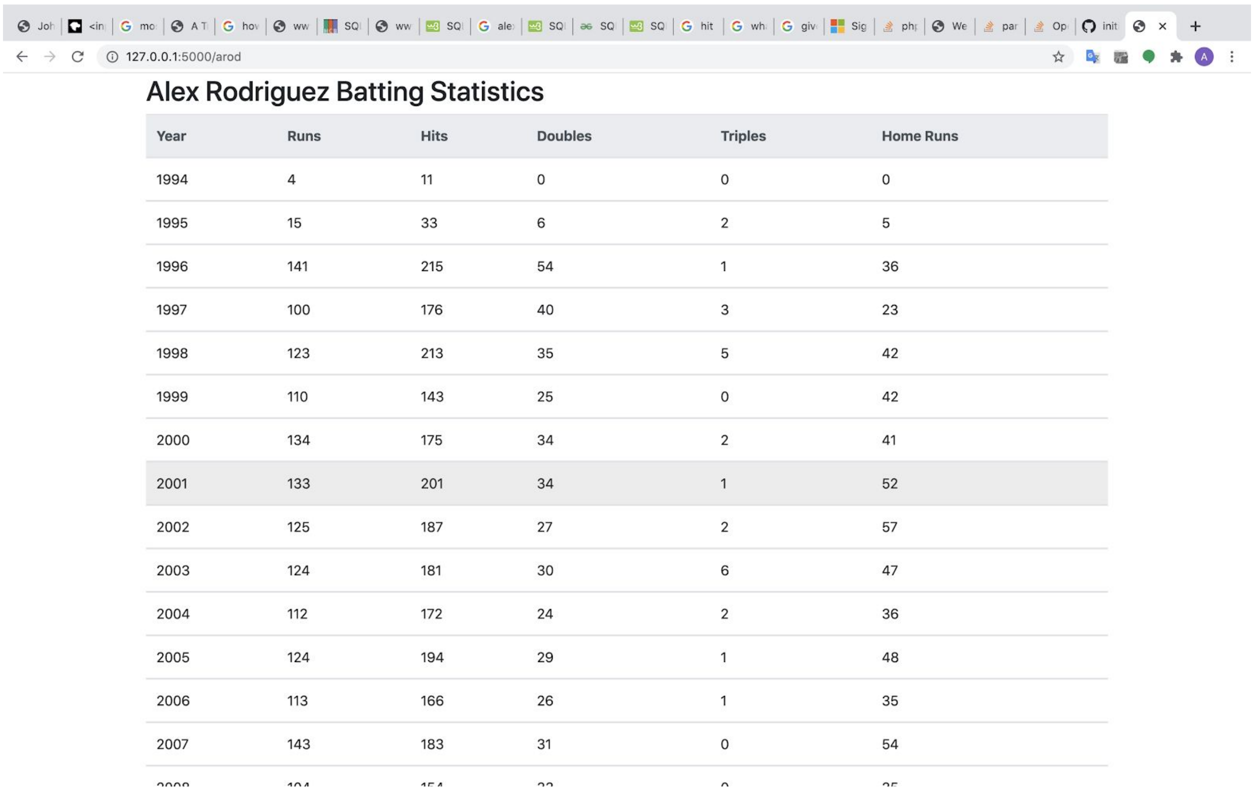
From here we can select Stats on Alex Rodriguez


127.0.0.1:5000/arod
Databases Fall 2020 One for the Ages!
Home
ARod Runs VS Average Batter
avgRuns
ARODRuns
140
120
100

A Tribute to Alex Rodriguez

We can see a picture of Alex Rodriguez along with a graph displaying his stats against league averages stats for the years of his career.

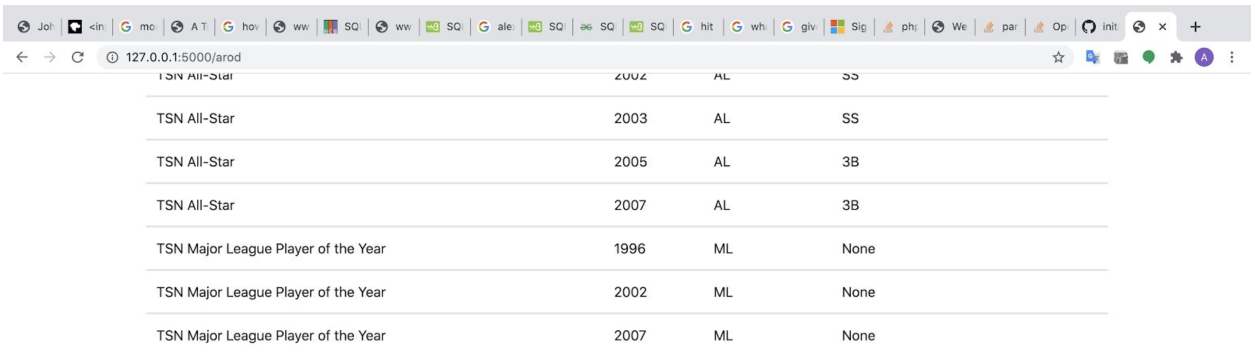




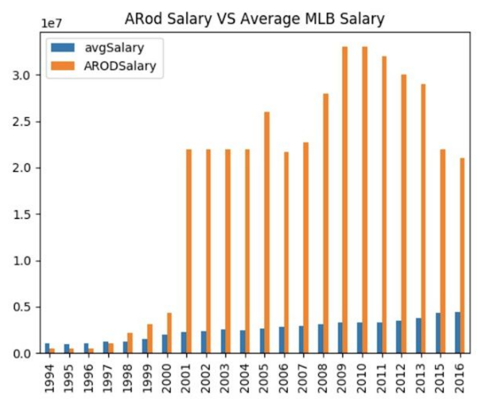
Alex Rodriguez Batting Statistics

Year	Runs	Hits	Doubles	Triples	Home Runs
1994	4	11	0	0	0
1995	15	33	6	2	5
1996	141	215	54	1	36
1997	100	176	40	3	23
1998	123	213	35	5	42
1999	110	143	25	0	42
2000	134	175	34	2	41
2001	133	201	34	1	52
2002	125	187	27	2	57
2003	124	181	30	6	47
2004	112	172	24	2	36
2005	124	194	29	1	48
2006	113	166	26	1	35
2007	143	183	31	0	54

We can see here a table with all of his stats laid out by the years of his career.

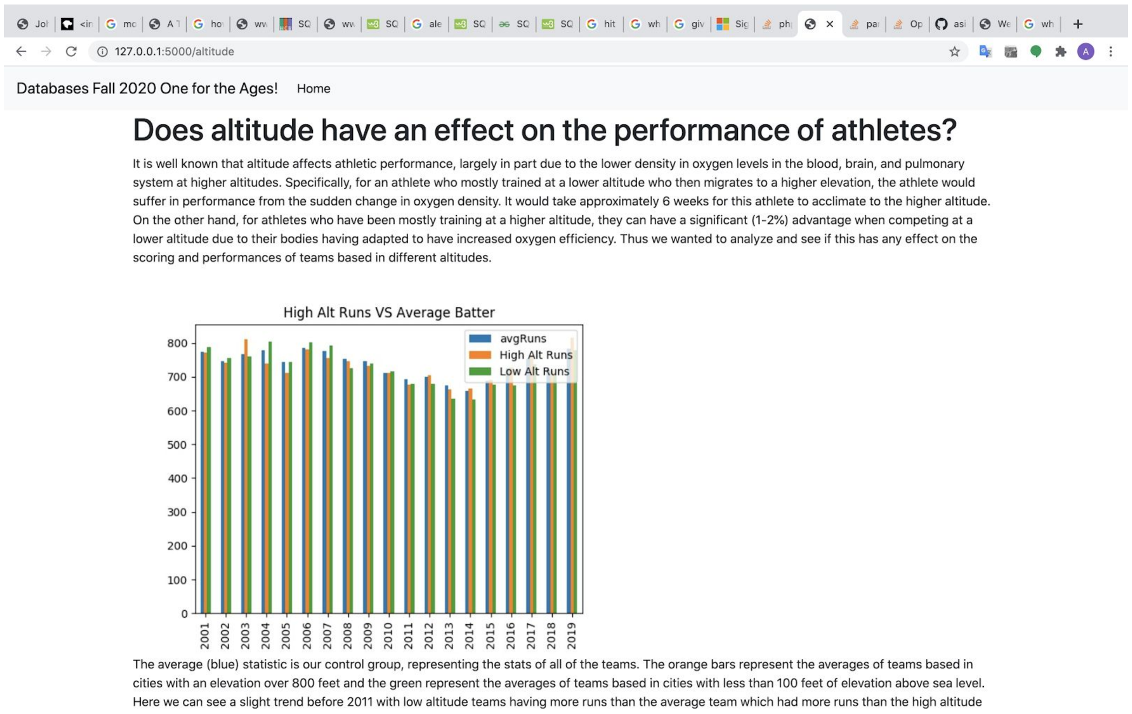


TSN All-Star	2002	AL	SS
TSN All-Star	2003	AL	SS
TSN All-Star	2005	AL	3B
TSN All-Star	2007	AL	3B
TSN Major League Player of the Year	1996	ML	None
TSN Major League Player of the Year	2002	ML	None
TSN Major League Player of the Year	2007	ML	None



We see a lot of different Alex Rodriguez statistics in this page.
From here we can click on the home button on the navigation bar on top.

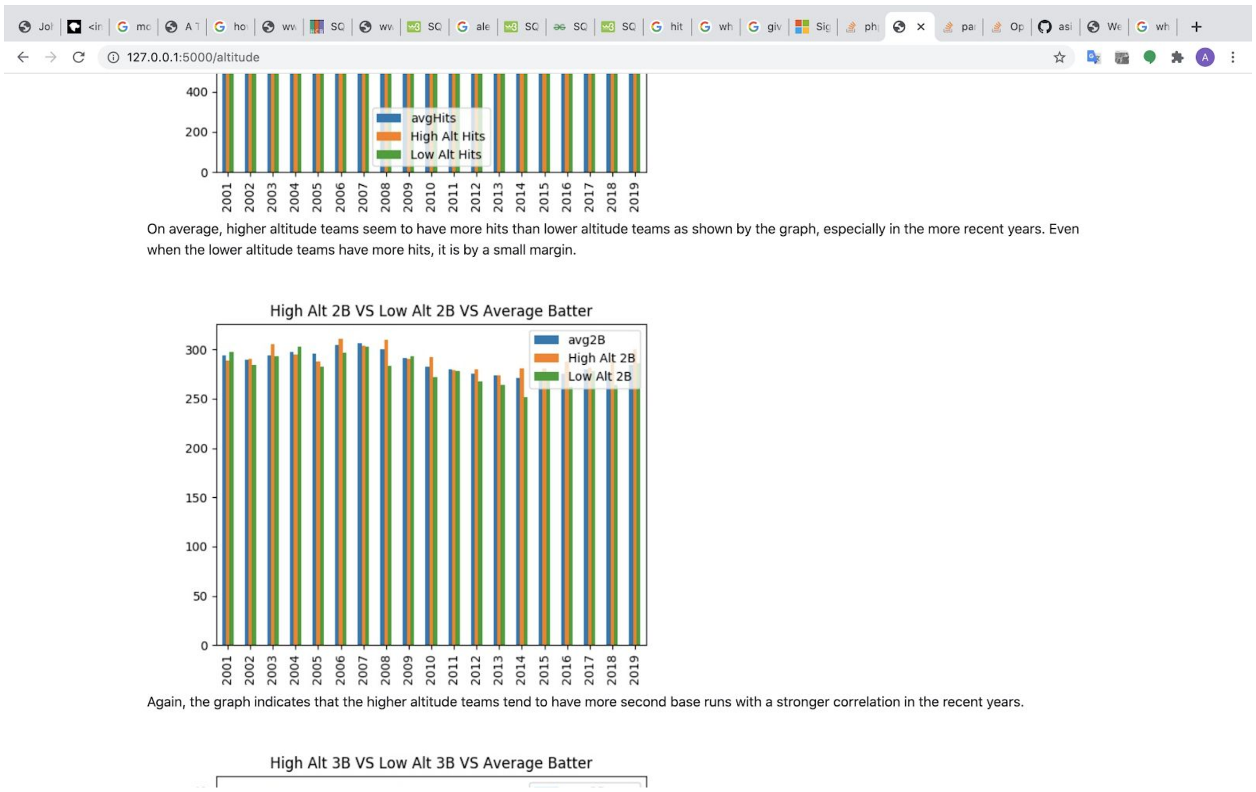
Once we get to the home page we can go to the Does Altitude have an Effect on the Performance of Athletes page.


127.0.0.1:5000/altitude
Databases Fall 2020 One for the Ages!
Home
High Alt Runs VS Average Batter
avgRuns
High Alt Runs
Low Alt Runs

Does altitude have an effect on the performance of athletes?

It is well known that altitude affects athletic performance, largely in part due to the lower density in oxygen levels in the blood, brain, and pulmonary system at higher altitudes. Specifically, for an athlete who mostly trained at a lower altitude who then migrates to a higher elevation, the athlete would suffer in performance from the sudden change in oxygen density. It would take approximately 6 weeks for this athlete to acclimate to the higher altitude. On the other hand, for athletes who have been mostly training at a higher altitude, they can have a significant (1-2%) advantage when competing at a lower altitude due to their bodies having adapted to have increased oxygen efficiency. Thus we wanted to analyze and see if this has any effect on the scoring and performances of teams based in different altitudes.

The average (blue) statistic is our control group, representing the stats of all of the teams. The orange bars represent the averages of teams based in cities with an elevation over 800 feet and the green represent the averages of teams based in cities with less than 100 feet of elevation above sea level. Here we can see a slight trend before 2011 with low altitude teams having more runs than the average team which had more runs than the high altitude

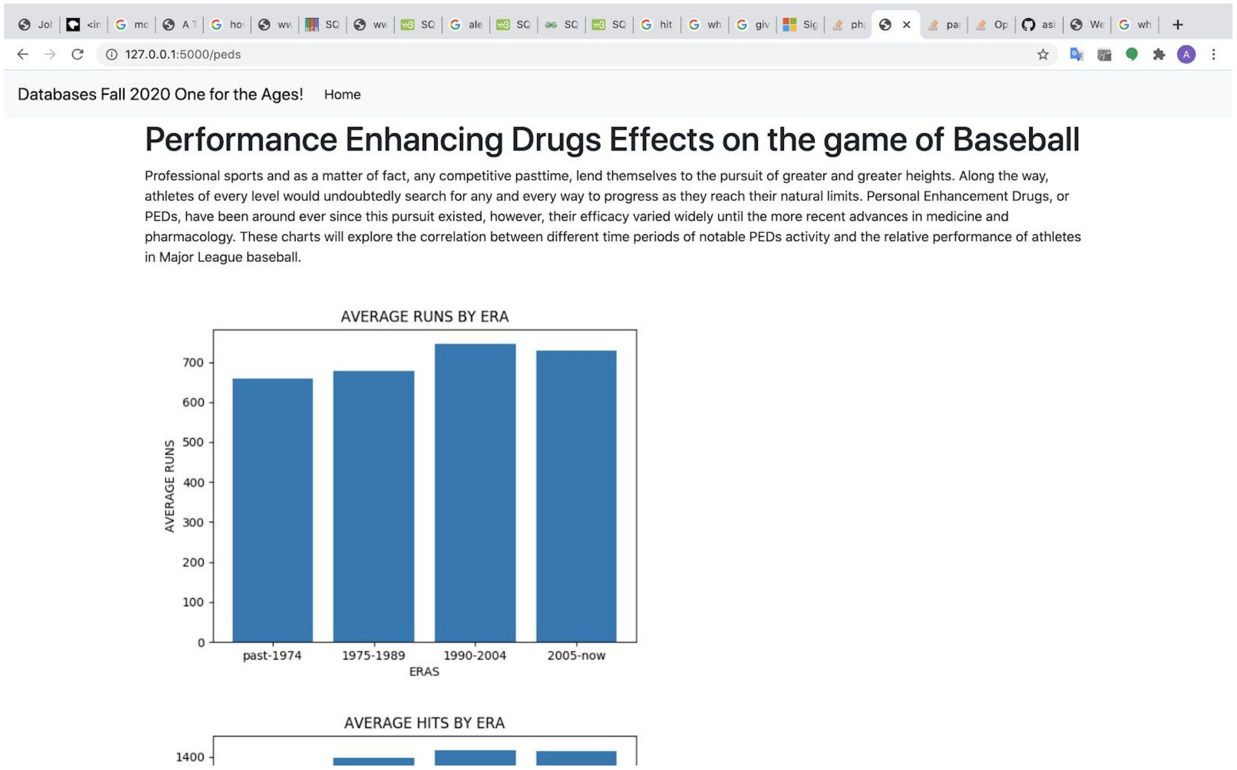

127.0.0.1:5000/altitude
avgHits
High Alt Hits
Low Alt Hits
High Alt 2B VS Low Alt 2B VS Average Batter
avg2B
High Alt 2B
Low Alt 2B
High Alt 3B VS Low Alt 3B VS Average Batter

On average, higher altitude teams seem to have more hits than lower altitude teams as shown by the graph, especially in the more recent years. Even when the lower altitude teams have more hits, it is by a small margin.

Again, the graph indicates that the higher altitude teams tend to have more second base runs with a stronger correlation in the recent years.

Here we can see graphs and discussion on the effects of altitude on the performance of athletes.
We can again go back to the home page from the navigation bar.

From here we can go to the The Effects of Performance Enhancing Drugs on the Game of Baseball page.

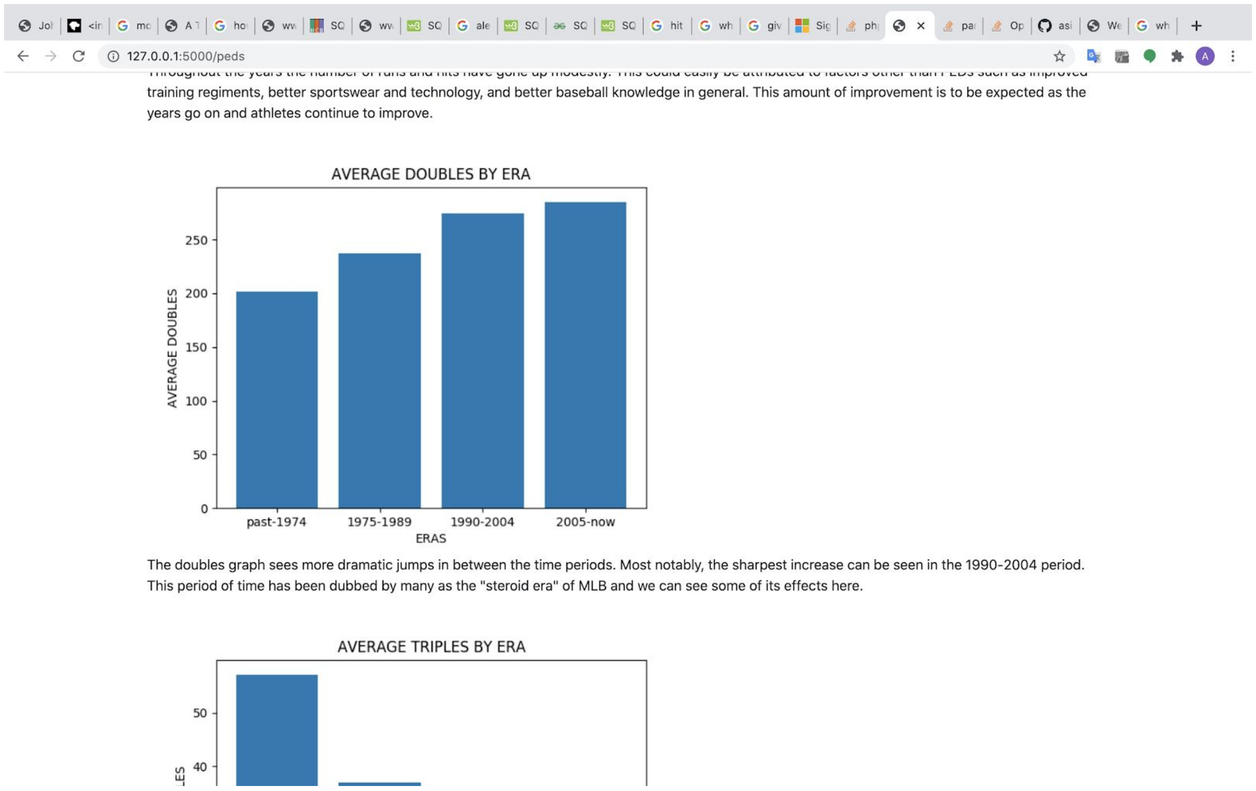


127.0.0.1:5000/peds

Databases Fall 2020 One for the Ages! Home

Performance Enhancing Drugs Effects on the game of Baseball

Professional sports and as a matter of fact, any competitive pasttime, lend themselves to the pursuit of greater and greater heights. Along the way, athletes of every level would undoubtedly search for any and every way to progress as they reach their natural limits. Personal Enhancement Drugs, or PEDs, have been around ever since this pursuit existed, however, their efficacy varied widely until the more recent advances in medicine and pharmacology. These charts will explore the correlation between different time periods of notable PEDs activity and the relative performance of athletes in Major League baseball.



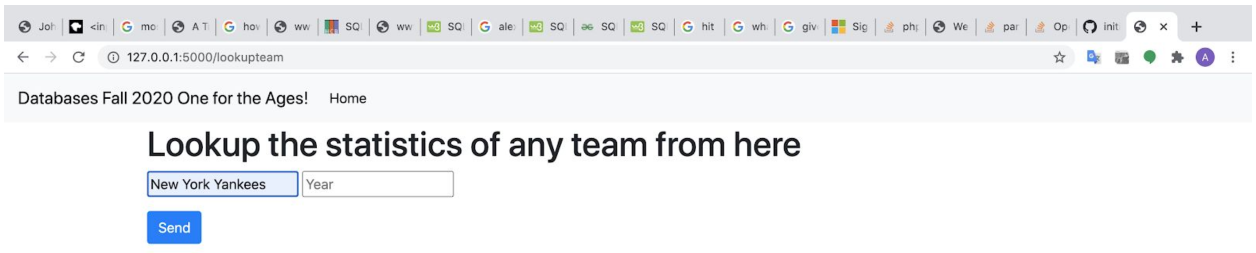
127.0.0.1:5000/peds

training regiments, better sportswear and technology, and better baseball knowledge in general. This amount of improvement is to be expected as the years go on and athletes continue to improve.

The doubles graph sees more dramatic jumps in between the time periods. Most notably, the sharpest increase can be seen in the 1990-2004 period. This period of time has been dubbed by many as the "steroid era" of MLB and we can see some of its effects here.

We see on this page graphs and discussion on the findings of the data.

We can go back to the homepage from the navigation bar on top and go to the look up team page.
Here we see that we can enter the team name and team year.
Here we can enter either the team name or the team name and year.
If we just enter New York Yankees like below



127.0.0.1:5000/lookupteam

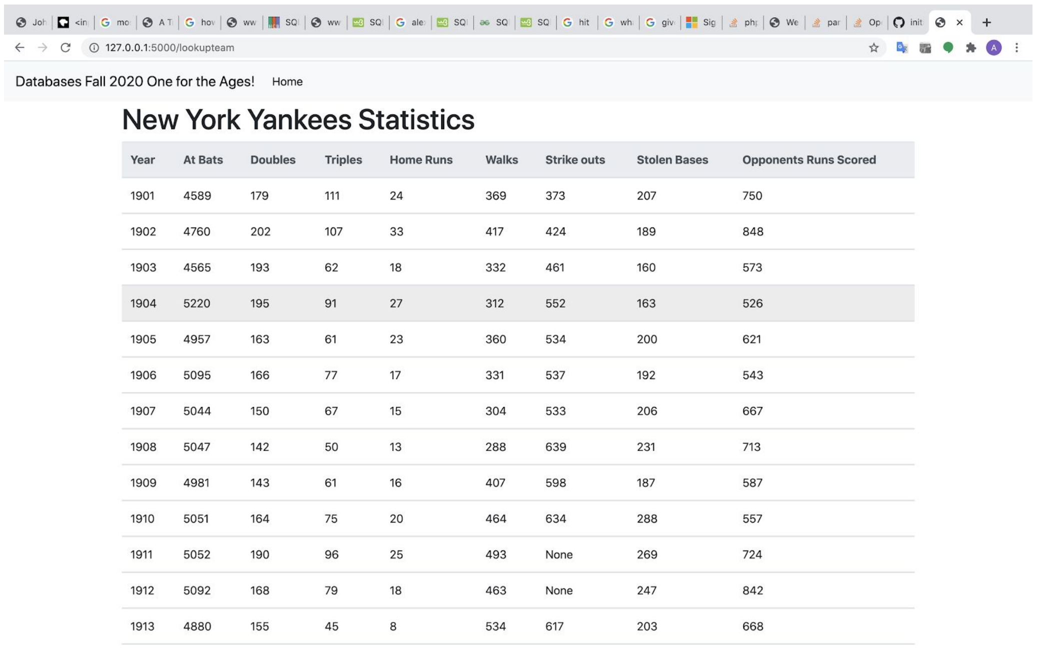
Databases Fall 2020 One for the Ages! Home

Lookup the statistics of any team from here

New York Yankees | Year

Send

From here we go to the page that displays the results for this query



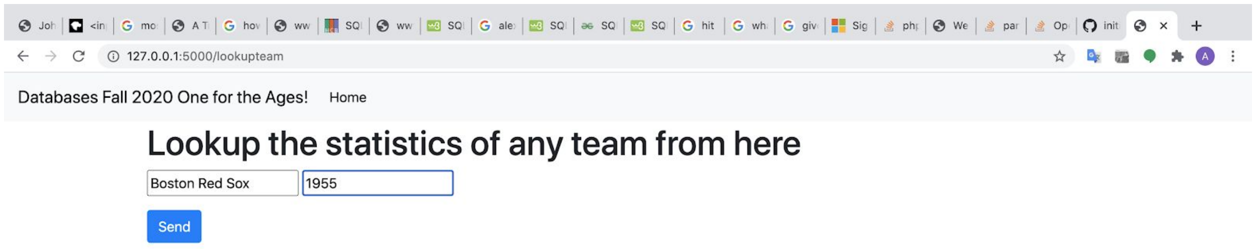
127.0.0.1:5000/lookupteam

Databases Fall 2020 One for the Ages! Home

New York Yankees Statistics

Year	At Bats	Doubles	Triples	Home Runs	Walks	Strike outs	Stolen Bases	Opponents Runs Scored
1901	4589	179	111	24	369	373	207	750
1902	4760	202	107	33	417	424	189	848
1903	4565	193	62	18	332	461	160	573
1904	5220	195	91	27	312	552	163	526
1905	4957	163	61	23	360	534	200	621
1906	5095	166	77	17	331	537	192	543
1907	5044	150	67	15	304	533	206	667
1908	5047	142	50	13	288	639	231	713
1909	4981	143	61	16	407	598	187	587
1910	5051	164	75	20	464	634	288	557
1911	5052	190	96	25	493	None	269	724
1912	5092	168	79	18	463	None	247	842
1913	4880	155	45	8	534	617	203	668

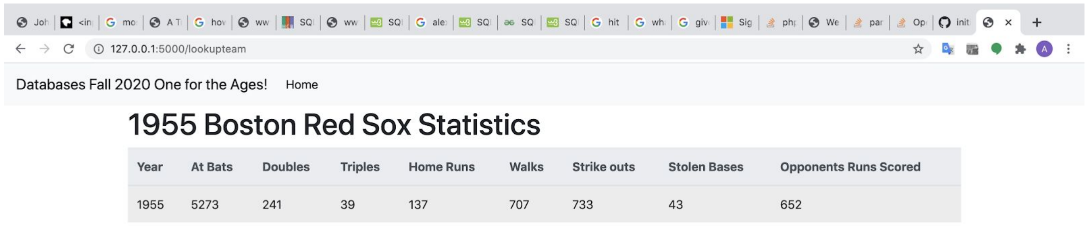
We can either go back using the back arrow or go to the home and go back to the search team page
This time we can search by team name and team year


127.0.0.1:5000/lookupteam
Databases Fall 2020 One for the Ages!
Home
Send

Lookup the statistics of any team from here

Boston Red Sox | 1955

From here we go to another page with the query results

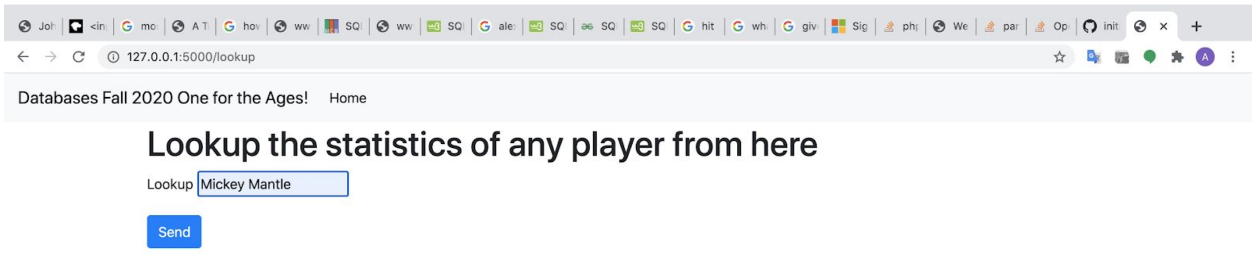

127.0.0.1:5000/lookupteam
Databases Fall 2020 One for the Ages!
Home

1955 Boston Red Sox Statistics

Year	At Bats	Doubles	Triples	Home Runs	Walks	Strike outs	Stolen Bases	Opponents Runs Scored
1955	5273	241	39	137	707	733	43	652

From here we can go back to the home page from the navigation bar

We can go to the search player page this time
We can input the name of a baseball player that played in MLB



127.0.0.1:5000/lookup

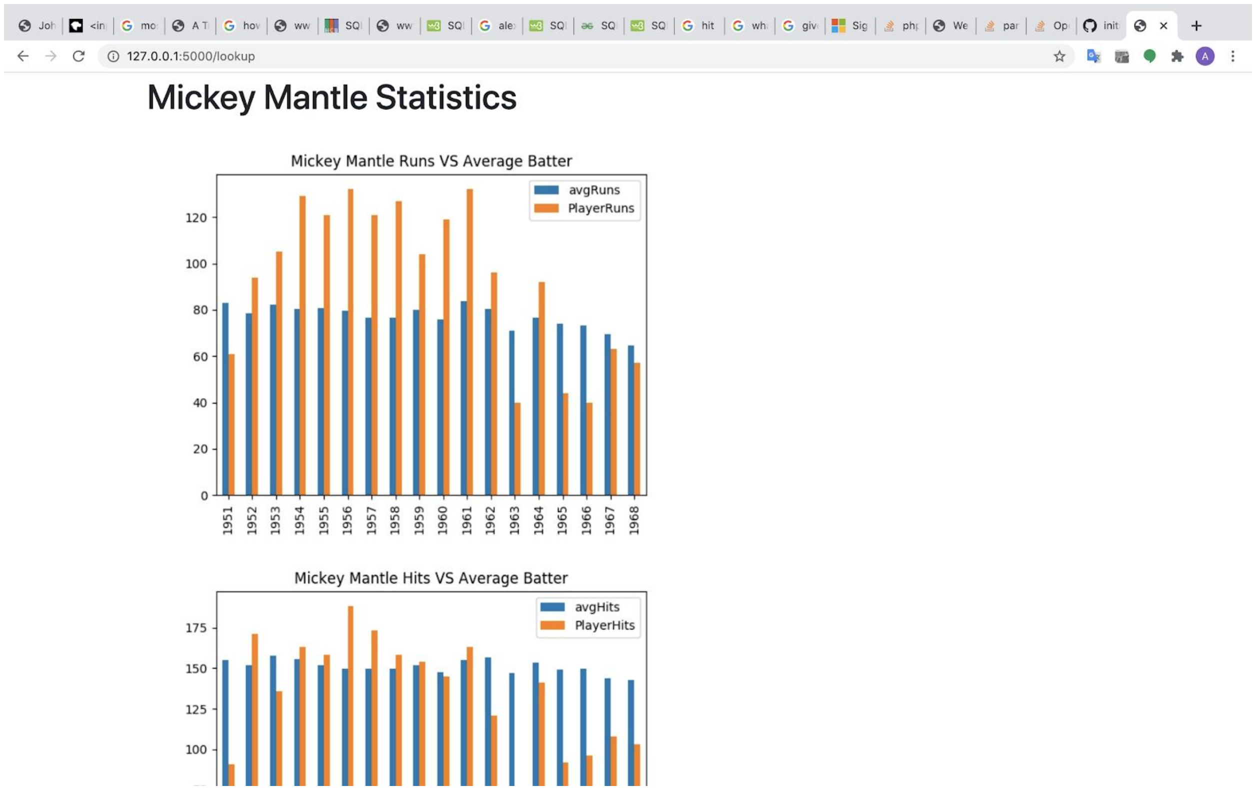
Databases Fall 2020 One for the Ages!   Home

Lookup the statistics of any player from here

Lookup Mickey Mantle

Send

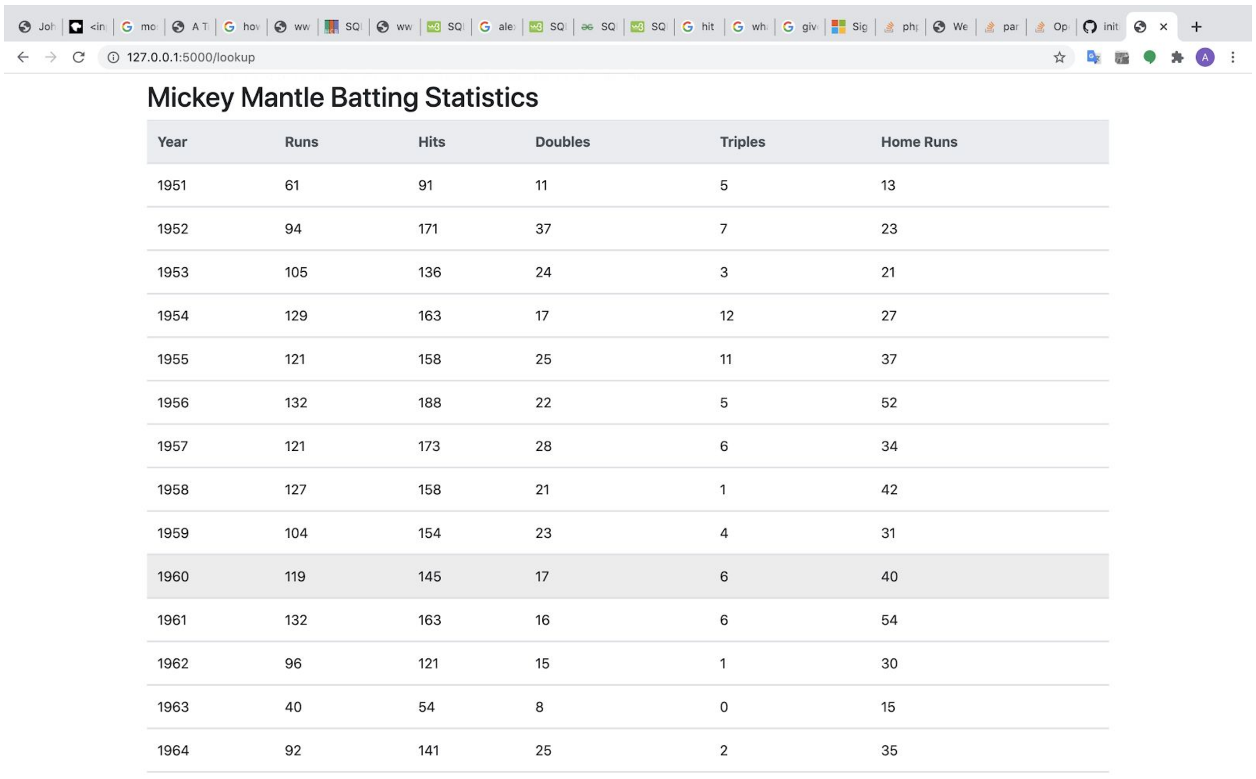
From here we are sent to another page with the query results



127.0.0.1:5000/lookup

Mickey Mantle Statistics

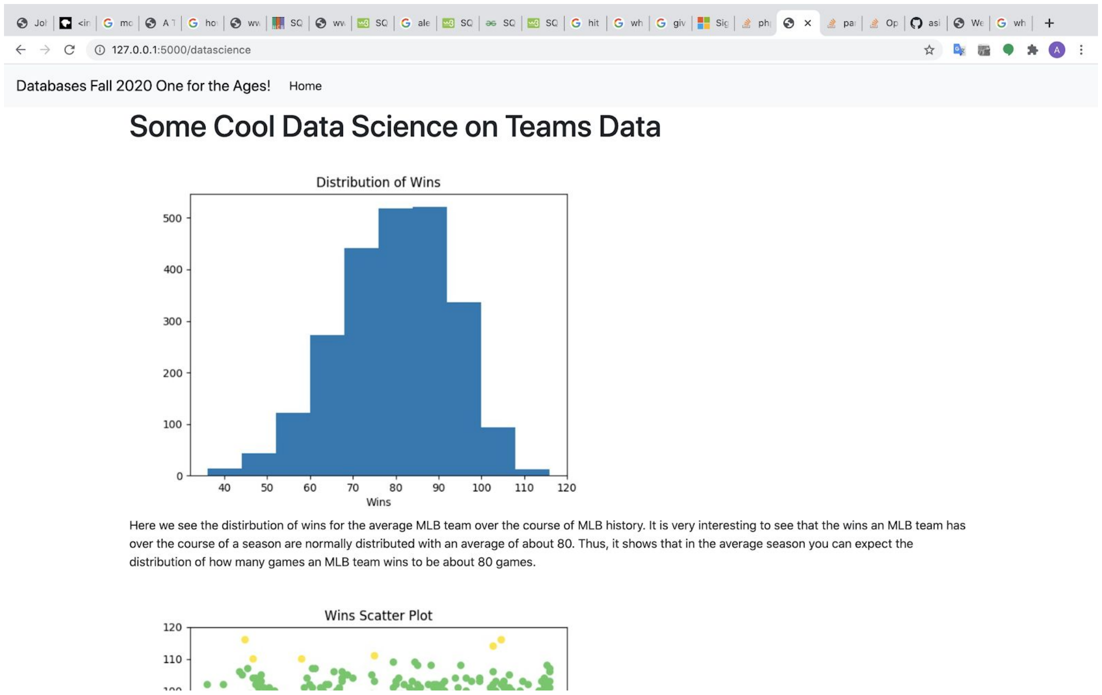
Here we see graphs and tables with the players statistics



Mickey Mantle Batting Statistics

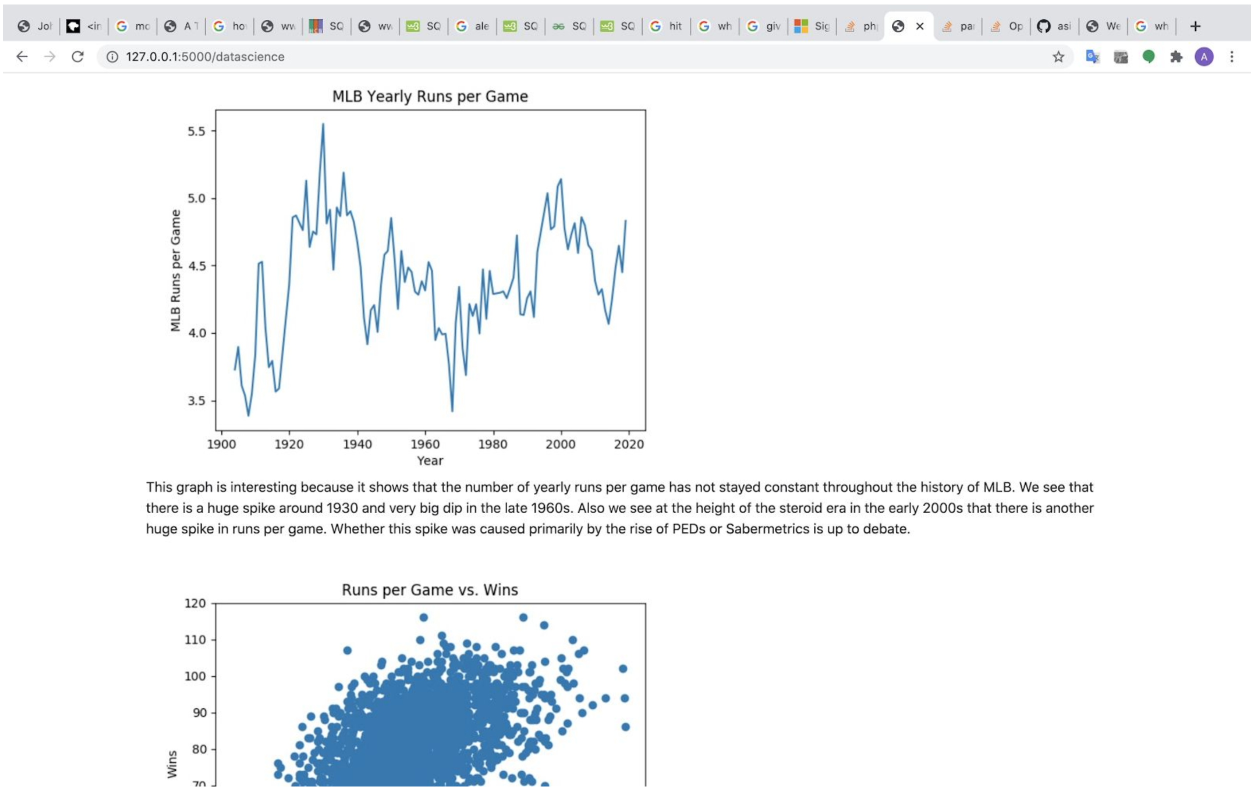
Year	Runs	Hits	Doubles	Triples	Home Runs
1951	61	91	11	5	13
1952	94	171	37	7	23
1953	105	136	24	3	21
1954	129	163	17	12	27
1955	121	158	25	11	37
1956	132	188	22	5	52
1957	121	173	28	6	34
1958	127	158	21	1	42
1959	104	154	23	4	31
1960	119	145	17	6	40
1961	132	163	16	6	54
1962	96	121	15	1	30
1963	40	54	8	0	15
1964	92	141	25	2	35

From here we can go back to the home page from the navigation bar
From the home page we can click on the cool Data Science page
On the cool data science page we see graphs and discussion implementing data science techniques on the game of baseball



Some Cool Data Science on Teams Data

Here we see the distirbution of wins for the average MLB team over the course of MLB history. It is very interesting to see that the wins an MLB team has over the course of a season are normally distributed with an average of about 80. Thus, it shows that in the average season you can expect the distribution of how many games an MLB team wins to be about 80 games.


127.0.0.1:5000/datascience

This graph is interesting because it shows that the number of yearly runs per game has not stayed constant throughout the history of MLB. We see that there is a huge spike around 1930 and very big dip in the late 1960s. Also we see at the height of the steroid era in the early 2000s that there is another huge spike in runs per game. Whether this spike was caused primarily by the rise of PEDs or Sabermetrics is up to debate.

This was the final page for the project and displayed different data science techniques you could use with baseball data.

(12)
<u>Tables</u>
Players
Managers
Batting (an entry per player per year)
Pitching (an entry per player per year)
Fielding (an entry per player per year)
Teams
AllstarFull

```
DROP TABLE IF EXISTS Players;
CREATE TABLE Players (
        playerID varchar(9) NOT NULL,
        birthYear int(11),
        birthMonth int(11),
        birthDay int(11),
        birthCountry varchar(255),
        birthState varchar(255),
        birthCity varchar(255),
        deathYear int(11),
        deathMonth int(11),
        deathDay int(11),
        deathCountry varchar(255),
        deathState varchar(255),
        deathCity varchar(255),
        nameFirst varchar(255),
        nameLast varchar(255),
        weight int(11),
        height int(11),
        bats varchar(255),
        throws varchar(255),
        PRIMARY KEY (playerID)
)
INSERT INTO Players VALUES
('aardsda01',1981,12,27,'USA','CO','Denver',NULL,NULL,NULL,NULL,NULL,NULL,'Davi
d','Aardsma'',215,75,'R','R')
INSERT INTO Players VALUES
('aaronto01',1939,8,5,'USA','AL','Mobile',1984,8,16,'USA','GA','Atlanta','Tommie','Aaron',
190,75,'R','R')



DROP TABLE IF EXISTS Managers;
CREATE TABLE Managers (
        playerID varchar(9),
        yearID smallint(6) NOT NULL,
        teamID char(3) NOT NULL,
        inseason smallint(6) NOT NULL,
        G smallint(6),
        W smallint(6),
        L smallint(6),
        teamRank smallint(6),
        plyrMgr varchar(1),
        PRIMARY KEY playerID (playerID, yearID),
        CONSTRAINT managers_fk FOREIGN KEY (teamID) REFERENCES teams
       (teamID),
)
INSERT INTO Managers VALUES ('wrighha01',1871,'BS1',1,31,20,10,3,'Y')
INSERT INTO Managers VALUES ('woodji01',1871,'CH1',1,28,19,9,2,'Y')


DROP TABLE IF EXISTS Teams;
CREATE TABLE Teams (
        yearID smallint(6) NOT NULL,
        teamID char(3) NOT NULL,
        teamRank smallint(6),
        G smallint(6),
        Ghome smallint(6),
        W smallint(6),
        L smallint(6),
        DivWin varchar(1),
        WCWin varchar(1),
        LgWin varchar(1),
        WSWin varchar(1),
        R smallint(6),
        AB smallint(6),
        H smallint(6),
        2B smallint(6),
        3B smallint(6),
        HR smallint(6),
        BB smallint(6),
        SO smallint(6),
        SB smallint(6),
        CS smallint(6),
        HBP smallint(6),
        SF smallint(6),
        RA smallint(6),
        ER smallint(6),
        ERA double,
        CG smallint(6),
        SHO smallint(6),
        SV smallint(6),
        IPouts int(11),
        HA smallint(6),
        HRA smallint(6),
        BBA smallint(6),
        SOA smallint(6),
        E int(11),
            DP int(11),
            FP double,
            name varchar(50),
            park varchar(255),
            attendance int(11),
            BPF int(11),
            PPF int(11),
            PRIMARY KEY teamID (teamID, yearID),
)
INSERT INTO Teams VALUES
(1871,'BS1',3,31,NULL,20,10,NULL,NULL,'N',NULL,401,1372,426,70,37,3,60,19,73,16,
NULL,NULL,303,109,3.55,22,1,3,828,367,2,42,23,243,24,0.8340000000000001,'Bosto
n Red Stockings','South End Grounds I',NULL,103,98)
INSERT INTO Teams VALUES
(1871,'CH1',2,28,NULL,19,9,NULL,NULL,'N',NULL,302,1196,323,52,21,10,60,22,69,21,
NULL,NULL,241,77,2.76,25,0,1,753,308,6,28,22,229,16,0.8290000000000001,'Chicag
o White Stockings','Union Base-Ball Grounds',NULL,104,102)



DROP TABLE IF EXISTS PlayerBatting;
CREATE TABLE PlayerBatting (
  playerID varchar(9) NOT NULL,
  yearID smallint(6) NOT NULL,
  stint smallint(6) NOT NULL,
  teamID char(3),
  G smallint(6),
  G_batting smallint(6),
  AB smallint(6),
  R smallint(6),
  H smallint(6),
  2B smallint(6),
  3B smallint(6),
  HR smallint(6),
  RBI smallint(6),
  SB smallint(6),
  CS smallint(6),
  BB smallint(6),
  SO smallint(6),
  IBB smallint(6),
  HBP smallint(6),
  SH smallint(6),
  SF smallint(6),
  GIDP smallint(6),
  PRIMARY KEY playerID (playerID, yearID, stint),
  CONSTRAINT batting_fk2 FOREIGN KEY (teamID) REFERENCES Teams (teamID),
  CONSTRAINT batting_fk2 FOREIGN KEY (playerID) REFERENCES Players
(playerID)
)
INSERT INTO PlayerBatting VALUES
('abercda01',1871,1,'TRO',1,NULL,4,0,0,0,0,0,0,0,0,0,0,NULL,NULL,NULL,NULL,0)
INSERT INTO PlayerBatting VALUES
('addybo01',1871,1,'RC1',25,NULL,118,30,32,6,0,0,13,8,1,4,0,NULL,NULL,NULL,NULL,
0)

DROP TABLE IF EXISTS PlayerPitching;
CREATE TABLE PlayerPitching (
  playerID varchar(9) NOT NULL,
  yearID smallint(6) NOT NULL,
  stint smallint(6) NOT NULL,
  teamID char(3),
  W smallint(6),
  L smallint(6),
  G smallint(6),
  GS smallint(6),
  CG smallint(6),
  SHO smallint(6),
  SV smallint(6),
  IPouts int(11),
  H smallint(6),
  ER smallint(6),
  HR smallint(6),
  BB smallint(6),
  SO smallint(6),
  BAOpp double,
  ERA double,
  IBB smallint(6),
  WP smallint(6),
  HBP smallint(6),
  BK smallint(6),
  BFP smallint(6),
  GF smallint(6),
  R smallint(6),
  SH smallint(6),
  SF smallint(6),
  GIDP smallint(6),
  PRIMARY KEY playerID (playerID,yearID,stint),
  CONSTRAINT pitching_fk1 FOREIGN KEY (teamID) REFERENCES Teams(teamID),
  CONSTRAINT pitching_fk2 FOREIGN KEY (playerID) REFERENCES
Players(playerID)
)

DROP TABLE IF EXISTS PlayerFielding;
CREATE TABLE PlayerFielding (
  playerID varchar(9) NOT NULL,
  yearID smallint(6) NOT NULL,
  stint smallint(6) NOT NULL,
  teamID char(3),
  lgID char(20),
  Pos char(3),
  G int(11),
  GS int(11),
  InnOuts int(11),
  PO int(6),
  A int(6),
  E int(6),
  DP int(6),
  PB int(6) only catchers,
  WP int(6) only catcher,
  SB int(6),
  CS int(6),
  ZR int(6)
  PRIMARY KEY playerID (playerID,yearID,stint),
  CONSTRAINT fielding_fk1 FOREIGN KEY (teamID) REFERENCES Teams(teamID),
  CONSTRAINT fielding_fk2 FOREIGN KEY (playerID) REFERENCES
Players(playerID)
)

DROP TABLE IF EXISTS AllstarFull;
```

```
CREATE TABLE AllstarFull (
  playerID varchar(9) NOT NULL,
  yearID smallint(6) NOT NULL,
  gamenum smallint(6) NOT NULL,
  teamID char(3),
  lgID char(20),
  Pos char(3),
  GP int(1),
  startingPos int(6)
  PRIMARY KEY playerID (playerID,yearID),
  CONSTRAINT allstar_fk1 FOREIGN KEY (teamID) REFERENCES Teams(teamID),
  CONSTRAINT allstar_fk2 FOREIGN KEY (playerID) REFERENCES Players(playerID)
)
```

(13)
SQL Code

```
  //Gets all information from People about player Alex Rodriguez
 posts = conn.execute('SELECT * FROM People WHERE nameFirst =
"Alex" AND nameLast = "Rodriguez"').fetchall()
 //Gets year,team,league,wins,and losses about Alex Rodriguez
manager
arodManager = conn.execute('SELECT yearID,teamID,lgID,W,L FROM
People INNER JOIN Managers ON People.playerID=Managers.playerID
WHERE People.nameFirst = "Alex" AND People.nameLast =
"Rodriguez"').fetchall()

//gets information about about Alex Rodriguez in the all star
game
arodAllStar = conn.execute('SELECT
YearID,teamID,lgID,startingPos FROM People INNER JOIN
AllstarFull ON People.playerID=AllstarFull.playerID WHERE
People.nameFirst = "Alex" AND People.nameLast =
"Rodriguez"').fetchall()
```

```
//Gets information on Alex Rodriguez batting statistics for
every year in the league
   arodBatting = conn.execute('SELECT
yearID,Batting.R,Batting.H,Batting."2B",Batting."3B",Batting.HR
FROM (People INNER JOIN Batting ON People.playerID =
Batting.playerID) WHERE People.nameFirst = "Alex" AND
People.nameLast = "Rodriguez"').fetchall()

//gets average batting statistics for every year Alex Rodriguez
was in the league and batters had more than 450 at bats
avgBatting = conn.execute('SELECT yearID,
AVG(Batting.R),AVG(Batting.H),AVG(Batting."2B"),AVG(Batting."3B"
),AVG(Batting.HR) FROM Batting WHERE AB > 450 AND yearID IN
(SELECT yearID FROM (People INNER JOIN Batting ON
People.playerID = Batting.playerID) WHERE People.nameFirst =
"Alex" AND People.nameLast = "Rodriguez") GROUP BY
yearID').fetchall()

//Gets fielding statistics for Alex Rodriguez
arodFielding = conn.execute('SELECT yearID,teamID,Pos,G,A,E,ZR
FROM People INNER JOIN Fielding ON
People.playerID=Fielding.playerID WHERE People.nameFirst =
"Alex" AND People.nameLast = "Rodriguez"').fetchall()

//Gets average zone rating for MLB every year ARod was in the
league
avgZoneR = conn.execute('SELECT yearID,AVG(ZR) FROM Fielding
WHERE yearID in (SELECT yearID FROM People INNER JOIN Fielding
ON People.playerID=Fielding.playerID WHERE People.nameFirst =
"Alex" AND People.nameLast = "Rodriguez") GROUP BY
yearID').fetchall()
```

```
//Gets year and salary for Alex Rodriguez
   arodSalary = conn.execute('SELECT yearID, salary FROM People
INNER JOIN Salaries ON People.playerID=Salaries.playerID WHERE
People.nameFirst = "Alex" AND People.nameLast =
"Rodriguez"').fetchall()

//Gets Average MLB salary statistics for every year Alex
Rodriguez was in the league
avgSalary = conn.execute('SELECT yearID, AVG(salary) FROM
SALARIES WHERE yearID IN (SELECT yearID FROM People INNER JOIN
Salaries ON People.playerID=Salaries.playerID WHERE
People.nameFirst = "Alex" AND People.nameLast = "Rodriguez")
GROUP BY yearID').fetchall()

//Gets all award statistics for Alex Rodriguez
arodAwards = conn.execute('SELECT * FROM People INNER JOIN
AwardsPlayers ON People.playerID=AwardsPlayers.playerID WHERE
People.nameFirst = "Alex" AND People.nameLast =
"Rodriguez"').fetchall()

//Gets average batting statistics for years above 2000 for high
altitude cities
highAltBatting = conn.execute('SELECT
yearID,AVG(Teams.R),AVG(Teams.H),AVG(Teams."2B"),AVG(Teams."3B")
,AVG(Teams.HR) FROM (Teams) WHERE (Teams.name = "Kansas City
Royals" OR Teams.name = "Minnesota Twins" OR Teams.name =
"Atlanta Braves" OR Teams.name = "Pittsburgh Pirates" OR
Teams.name = "Arizona Diamondbacks" OR Teams.name = "Colorado
Rockies") AND yearID > 2000 GROUP BY yearID').fetchall()

//Gets average batting statistics for years above 2000 for low
altitude cities
```

```
 lowAltBatting = conn.execute('SELECT
yearID,AVG(Teams.R),AVG(Teams.H),AVG(Teams."2B"),AVG(Teams."3B")
,AVG(Teams.HR) FROM (Teams) WHERE (Teams.name = "New York
Yankees" OR Teams.name = "Tampa Bay Rays" OR Teams.name =
"Houston Astros" OR Teams.name = "Oakland Athletics" OR
Teams.name = "Miami Marlins" OR Teams.name = "New York Mets" OR
Teams.name = "Philadelphia Phillies" OR Teams.name = "San Diego
Padres" OR Teams.name = "San Francisco Giants") AND yearID >
2000 GROUP BY yearID').fetchall()

//Gets average batting statistics for years above 2000 for all
cities
avgBatting = conn.execute('SELECT
yearID,AVG(Teams.R),AVG(Teams.H),AVG(Teams."2B"),AVG(Teams."3B")
,AVG(Teams.HR) FROM (Teams) WHERE yearID > 2000 GROUP BY
yearID').fetchall()

//Gets average batting statistics for the era of pre 1975
baseball
pre1975Batting = conn.execute('SELECT
AVG(Teams.R),AVG(Teams.H),AVG(Teams."2B"),AVG(Teams."3B"),AVG(Te
ams.HR) FROM (Teams) WHERE yearID < 1975').fetchall()

//Gets average batting statistics for 1975-1990 era baseball
from75to90Batting = conn.execute('SELECT
AVG(Teams.R),AVG(Teams.H),AVG(Teams."2B"),AVG(Teams."3B"),AVG(Te
ams.HR) FROM (Teams) WHERE yearID >= 1975 AND yearID <
1990').fetchall()

//Gets average batting statistics for 1990-2005 “Steroid” era
baseball
```

```
from90to05Batting = conn.execute('SELECT
AVG(Teams.R),AVG(Teams.H),AVG(Teams."2B"),AVG(Teams."3B"),AVG(Te
ams.HR) FROM (Teams) WHERE yearID >= 1990 AND yearID <
2005').fetchall()

//Gets average batting statistics for post 2005 modern era
baseball
post2005Batting = conn.execute('SELECT
AVG(Teams.R),AVG(Teams.H),AVG(Teams."2B"),AVG(Teams."3B"),AVG(Te
ams.HR) FROM (Teams) WHERE yearID >= 2005').fetchall()

//Gets all team and franchise statistics for all teams
 Teams = conn.execute('select
yearID,lgID,teamID,Teams.franchID,divID,G,Ghome,W,L,DivWin,WCWin
,LgWin,WSWin,R,AB,H,"2B","3B",HR,BB,SO,SB,CS,HBP,SF,RA,ER,ERA,CG
,SHO,SV,IPouts,HA,HRA,BBA,SOA,E,DP,FP,name,park,attendance,BPF,P
PF,teamIDBR,teamIDlahman45,teamIDretro,TeamsFranchises.franchID,
franchName,active,NAassoc from Teams inner join TeamsFranchises
on Teams.franchID == TeamsFranchises.franchID where Teams.G >=
150 and TeamsFranchises.active == "Y"').fetchall()

//Gets statistics for all teams with a certain name given as
input
getTeam = conn.execute('SELECT
yearID,W,L,R,AB,H,"2B","3B",HR,BB,SO,SB,RA FROM Teams INNER JOIN
TeamsFranchises ON Teams.franchID=TeamsFranchises.franchID WHERE
TeamsFranchises.franchName = ' + name).fetchall()

// Gets statistics for all teams with a certain name and year
given as input
getTeam = conn.execute('SELECT
yearID,W,L,R,AB,H,"2B","3B",HR,BB,SO,SB,RA FROM Teams INNER JOIN
```

```
TeamsFranchises ON Teams.franchID=TeamsFranchises.franchID WHERE
TeamsFranchises.franchName = ' + name + ' AND Teams.yearID = ' +
year).fetchall()

//Gets statistics for average team in a certain year given as
input
avgTeam = conn.execute('SELECT
AVG(W),AVG(L),AVG(R),AVG(AB),AVG(H),AVG("2B"),AVG("3B"),AVG(HR),
AVG(BB),AVG(SO),AVG(SB),AVG(RA) FROM Teams INNER JOIN
TeamsFranchises ON Teams.franchID=TeamsFranchises.franchID WHERE
Teams.yearID = ' + year).fetchall()

//Gets manager statistics for a certain player name given as
input
playerManager = conn.execute('SELECT yearID,teamID,lgID,W,L FROM
People INNER JOIN Managers ON People.playerID=Managers.playerID
WHERE People.nameFirst = ' + firstName + ' AND People.nameLast =
' + lastName).fetchall()

//Gets all star statistics for a certain player name given as
input
playerAllStar = conn.execute('SELECT
YearID,teamID,lgID,startingPos FROM People INNER JOIN
AllstarFull ON People.playerID=AllstarFull.playerID WHERE
People.nameFirst = ' + firstName + ' AND People.nameLast = ' +
lastName).fetchall()

//gets batting statistics for a certain player name given as
input
   playerBatting = conn.execute('SELECT
yearID,Batting.R,Batting.H,Batting."2B",Batting."3B",Batting.HR
FROM (People INNER JOIN Batting ON People.playerID =
Batting.playerID) WHERE People.nameFirst = ' + firstName + ' AND
People.nameLast = ' + lastName).fetchall()

//Gets average batting statistics for a certain player name
given as input
avgBatting = conn.execute('SELECT yearID,
AVG(Batting.R),AVG(Batting.H),AVG(Batting."2B"),AVG(Batting."3B"
),AVG(Batting.HR) FROM Batting WHERE AB > 450 AND yearID IN
(SELECT yearID FROM (People INNER JOIN Batting ON
People.playerID = Batting.playerID) WHERE People.nameFirst = ' +
firstName + ' AND People.nameLast = ' + lastName + ') GROUP BY
yearID').fetchall()

//Gets fielding statistics for a certain player name given as
input
playerFielding = conn.execute('SELECT yearID,teamID,Pos,G,A,E,ZR
FROM People INNER JOIN Fielding ON
People.playerID=Fielding.playerID WHERE People.nameFirst = ' +
firstName + ' AND People.nameLast = ' + lastName).fetchall()

//Gets average zone rating for a certain player name given as
input
avgZoneR = conn.execute('SELECT yearID,AVG(ZR) FROM Fielding
WHERE yearID in (SELECT yearID FROM People INNER JOIN Fielding
ON People.playerID=Fielding.playerID WHERE People.nameFirst = '
+ firstName + ' AND People.nameLast = ' + lastName + ') GROUP BY
yearID').fetchall()

//Gets award statistics for a certain player name given as input
   playerAwards = conn.execute('SELECT * FROM People INNER JOIN
AwardsPlayers ON People.playerID=AwardsPlayers.playerID WHERE
```

```
People.nameFirst = ' + firstName + ' AND People.nameLast = ' +
lastName).fetchall()
```

Data input code

```
//takes in text input for team name and year from user
<form action="/lookupteam" method="POST">
   <div class="form-group">
       <input type="text" name="name" placeholder="Name"
id="name">
       <input type="text" name="year" placeholder="Year"
id="year">
   </div>
   <input class="btn btn btn-primary" type="submit"
value="Send"/>
 </form>

//takes in text input for player name from user
<form action="/lookup" method="POST">
   <div class="form-group">
       <label for="name">Lookup</label>
       <input type="text" name="name" placeholder="First and
Last Name" id="name">
   </div>
   <input class="btn btn btn-primary" type="submit"
value="Send"/>
 </form>
```

(1) Databases Final Project Phase 1

Databases Final Project
Phase I

(1) Who are your team members: Stanley Zheng, Amritpal Singh

(2) Briefly describe your target domain (e.g. a world geographic database) :
Baseball Databases

- Players, Player Statistics, Managers, Teams, Team statistics, team schedules

(3) Give a reasonably comprehensive and representative list of the kinds of English questions you would like your system to be able to answer (minimum 15). For example, "Compute the mean literacy rate for countries with a per capita income of less than $400/year, grouped by continent." Please note that these queries are not the only thing you will need to support, just some basic objectives to help focus your design choices.

1. Which players have played for a team with a player that played for a team that Yogi Berra played for?
2. Which team has the highest average home runs per game and what is that average?
3. What is the average number of strikeouts per game in 1998, grouped by teams with less than 10 wins that year?
4. List the players with more wins than losses on their record and has played for only 1 team.
5. Of the players who are still alive, how many players were born in the same state as where another player died?
6. List the manager(s) who have managed the most distinct players that bat left handed no matter which team they were on.
7. List the games played in 1963 in Maryland that ended in a tie as well as the teams that played those games.
8. Compute the mean salary between the years 1990-2000 for players whose last name is Smith, grouped by teams.
9. List the players and their birth cities who have played in and won a game in their birth city.
10. List the teams and their managers at the time of the teams who have won the World Series in the same year where they lost more games than they won.
11. How many total strikeouts have the Baltimore Canaries thrown across all their games, grouped by managers?
12. List the players who played on the same team as a player who later became a manager.

13. List the players that have won in a game against a team Yogi Berra was on and also won a game against a team Jackie Robinson was on.
14. List the players who have played in every position at least once.
15. How many games were played in Pennsylvania in which the home team lost?
16. How many teams have won more than 5 World Series?
17. What team won the World Series in 1967?
18. List the teams that Babe Ruth has played on.
19. List all players with more than 10 home runs during the 1954 season.
20. List all players who played with Babe Ruth who had more than 10 home runs during their season with Babe Ruth.

Some Questions we want done by end of week:
Players Questions

- Who has ever played with a certain *player (fname, lname)*

    -
      ```
      SELECT p1.nameFirst, p1.nameLast
      FROM Players p1, Players p2, PlaysFor pf1, PlaysFor pf2
      WHERE p1.retroID = pf1.retroID AND
              p2.retroID = pf2.retroID AND
              pf1.teamID = pf2.teamID AND
              p2.nameFirst = fname AND
              p2.nameLast = lname
      ```
- Players with at least a certain amount of *homeruns*, *triples*, *doubles...*

    -
      ```
      SELECT p.nameFirst, p.nameLast
      FROM Players p, Batting b
      WHERE p.playerID = b.playerID AND
              b.HR >= homeruns AND
              b.3B >= triples AND
              b.2B >= doubles
      (optional conditionals depending on which fields are inputted)
      ```
- Stats for a certain *player(fname, name)* with multiple bar graphs, one for each year

    -
      ```
      SELECT b.yearID, b.teamID, b.lgID, b.G, b.R, b.H, b.2B, b.3B, b.HR
      (more stats if you want)
      FROM Players p, Batting b
      WHERE p.nameFirst = fname AND
              p.nameLast = lname AND
              p.playerID = b.playerID
      ```
- Birth city/state/country for *player (fname, lname)*

    -
      ```
      SELECT p.birthCity, p.birthState, p.birthCountry
      FROM Players p
      ```

```
        WHERE p.nameFirst = fname AND
              p.nameLast = lname
```
- Players with a certain *hit percentage*
- Player search by *isAlive, batting hand, birthplace* (these fields are optional)

    -
      ```
      SELECT p.nameFirst, p.nameLast
      FROM Players p
      WHERE p.bats =
      ```
- List all teams a *player* has played on

Teams Questions

- How many games did a *team* win in a certain *year*
- Team wins by *year* with graph
- Members of team by *year*
- Team above certain *average hit percentage*

(4) Design and show a relational data model that you plan to use for your system, with a preliminary implementation in standard SQL data-definition-language syntax. This specification should include appropriate primary key, foreign key and domain specifications for each relation/attribute, as well as the not null constraint when appropriate. You may also find it useful, but not required, to create a few insert-into statements that populate your schema designs with representative values (both to document your choices and to exercise them. You are welcome to change and augment your design and its specification by Phase II, but any time investment now will reduce effort later.

Tables
Players
Managers
Batting (an entry per player per year)
Pitching (an entry per player per year)
Fielding (an entry per player per year)
PlaysFor
Games

```
DROP TABLE IF EXISTS Players;
CREATE TABLE Players (
        playerID varchar(9) NOT NULL,
        birthYear int(11),
        birthMonth int(11),
```

```
	birthDay int(11),
	birthCountry varchar(255),
	birthState varchar(255),
	birthCity varchar(255),
	deathYear int(11),
	deathMonth int(11),
	deathDay int(11),
	deathCountry varchar(255),
	deathState varchar(255),
	deathCity varchar(255),
	nameFirst varchar(255),
	nameLast varchar(255),
	weight int(11),
	height int(11),
	bats varchar(255),
	throws varchar(255),
	PRIMARY KEY (playerID)
)
INSERT INTO Players VALUES
('aardsda01',1981,12,27,'USA','CO','Denver',NULL,NULL,NULL,NULL,NULL,NULL,'Davi
d','Aardsma'',215,75,'R','R')
INSERT INTO Players VALUES
('aaronto01',1939,8,5,'USA','AL','Mobile',1984,8,16,'USA','GA','Atlanta','Tommie','Aaron',
190,75,'R','R')



DROP TABLE IF EXISTS Managers;
CREATE TABLE Managers (
	playerID varchar(9),
	yearID smallint(6) NOT NULL,
	teamID char(3) NOT NULL,
	inseason smallint(6) NOT NULL,
	G smallint(6),
	W smallint(6),
	L smallint(6),
	teamRank smallint(6),
	plyrMgr varchar(1),
	PRIMARY KEY playerID (playerID, yearID),
            CONSTRAINT managers_fk FOREIGN KEY (teamID) REFERENCES teams
          (teamID),
)
INSERT INTO Managers VALUES ('wrighha01',1871,'BS1',1,31,20,10,3,'Y')
INSERT INTO Managers VALUES ('woodji01',1871,'CH1',1,28,19,9,2,'Y')


DROP TABLE IF EXISTS Teams;
CREATE TABLE Teams (
            yearID smallint(6) NOT NULL,
            teamID char(3) NOT NULL,
            teamRank smallint(6),
            G smallint(6),
            Ghome smallint(6),
            W smallint(6),
            L smallint(6),
            DivWin varchar(1),
            WCWin varchar(1),
            LgWin varchar(1),
            WSWin varchar(1),
            R smallint(6),
            AB smallint(6),
            H smallint(6),
            2B smallint(6),
            3B smallint(6),
            HR smallint(6),
            BB smallint(6),
            SO smallint(6),
            SB smallint(6),
            CS smallint(6),
            HBP smallint(6),
            SF smallint(6),
            RA smallint(6),
            ER smallint(6),
            ERA double,
            CG smallint(6),
            SHO smallint(6),
            SV smallint(6),
            IPouts int(11),
            HA smallint(6),
            HRA smallint(6),
            BBA smallint(6),
            SOA smallint(6),
            E int(11),
            DP int(11),
            FP double,
            name varchar(50),
            park varchar(255),
            attendance int(11),
            BPF int(11),
            PPF int(11),
            PRIMARY KEY teamID (teamID, yearID),
)
INSERT INTO Teams VALUES
(1871,'BS1',3,31,NULL,20,10,NULL,NULL,'N',NULL,401,1372,426,70,37,3,60,19,73,16,
NULL,NULL,303,109,3.55,22,1,3,828,367,2,42,23,243,24,0.8340000000000001,'Bosto
n Red Stockings','South End Grounds I',NULL,103,98)
INSERT INTO Teams VALUES
(1871,'CH1',2,28,NULL,19,9,NULL,NULL,'N',NULL,302,1196,323,52,21,10,60,22,69,21,
NULL,NULL,241,77,2.76,25,0,1,753,308,6,28,22,229,16,0.8290000000000001,'Chicag
o White Stockings','Union Base-Ball Grounds',NULL,104,102)



DROP TABLE IF EXISTS PlayerBatting;
CREATE TABLE PlayerBatting (
  playerID varchar(9) NOT NULL,
  yearID smallint(6) NOT NULL,
  stint smallint(6) NOT NULL,
  teamID char(3),
  G smallint(6),
  G_batting smallint(6),
  AB smallint(6),
  R smallint(6),
  H smallint(6),
  2B smallint(6),
  3B smallint(6),
  HR smallint(6),
  RBI smallint(6),
  SB smallint(6),
  CS smallint(6),
  BB smallint(6),
  SO smallint(6),
  IBB smallint(6),
  HBP smallint(6),
  SH smallint(6),
  SF smallint(6),
  GIDP smallint(6),
  PRIMARY KEY playerID (playerID, yearID, stint),
  CONSTRAINT batting_fk2 FOREIGN KEY (teamID) REFERENCES Teams (teamID),
  CONSTRAINT batting_fk2 FOREIGN KEY (playerID) REFERENCES Players
(playerID)
)
INSERT INTO PlayerBatting VALUES
('abercda01',1871,1,'TRO',1,NULL,4,0,0,0,0,0,0,0,0,0,0,NULL,NULL,NULL,NULL,0)
INSERT INTO PlayerBatting VALUES
('addybo01',1871,1,'RC1',25,NULL,118,30,32,6,0,0,13,8,1,4,0,NULL,NULL,NULL,NULL,
0)
DROP TABLE IF EXISTS PlayerPitching;
CREATE TABLE PlayerPitching (
  playerID varchar(9) NOT NULL,
  yearID smallint(6) NOT NULL,
  stint smallint(6) NOT NULL,
  teamID char(3),
  W smallint(6),
  L smallint(6),
  G smallint(6),
  GS smallint(6),
  CG smallint(6),
  SHO smallint(6),
  SV smallint(6),
  IPouts int(11),
  H smallint(6),
  ER smallint(6),
  HR smallint(6),
  BB smallint(6),
  SO smallint(6),
  BAOpp double,
  ERA double,
  IBB smallint(6),
  WP smallint(6),
  HBP smallint(6),
  BK smallint(6),
  BFP smallint(6),
  GF smallint(6),
  R smallint(6),
  SH smallint(6),
  SF smallint(6),
  GIDP smallint(6),
  PRIMARY KEY playerID (playerID,yearID,stint),
  CONSTRAINT pitching_fk1 FOREIGN KEY (teamID) REFERENCES Teams(teamID),
  CONSTRAINT pitching_fk2 FOREIGN KEY (playerID) REFERENCES
Players(playerID)
)
INSERT INTO PlayerPitching VALUES
('bechtge01',1871,1,'PH1',1,2,3,3,2,0,0,78,43,23,0,11,1,NULL,7.96,NULL,7,NULL,0,146
,0,42,NULL,NULL,NULL)
```

(5) Submit a set of SQL statements that will implement a representative sample of your target queries, including some of the more interesting or challenging cases. This is primarily to get you to think about your design and how it will be exercised as well as any limitations, so focus on queries that would be useful for doing so, rather than creating trivial or non-insightful queries just to fill space.

- What is the average number of strikeouts in 1998, grouped by teams with less than 10 wins that year?

    -
      ```
      SELECT t.teamID, AVG(t.SO)
      FROM Teams t
      WHERE t.yearID = 1998 AND t.W < 10
      GROUP BY t.teamID
      ```
- Which players have played for a team with a player that played for a team that Yogi Berra played for?

    -
      ```
      SELECT p1.nameFirst, p1.nameLast
      FROM Players p1, Players p2, Players yb, PlaysFor pf1, PlaysFor pf2,
      PlaysFor pf3, PlaysFor pf4
      WHERE pf1.playerID = p1.playerID AND
      ```

    pf2.playerID = p2.playerID AND
    pf1.team = pf2.team AND
    pf3.playerID = p2.playerID AND
    pf4.playerID = yb.playerID AND
    pf3.team = pf4.team

- How many teams have won more than 5 World Series?
    - SELECT WSWin
      FROM Teams
      WHERE WSWin > 5
      GROUP BY teamID
- What team won the World Series in 1967?
    - SELECT name
      FROM Teams
      WHERE yearID = 1967
- List the teams that Babe Ruth has played on.
    - SELECT t.name
      FROM Teams AS t, Players AS p, PlayerBatting AS pb
      WHERE p.playerid = pb.playerid AND t.teamid = pb.teamid AND
      p.nameFirst = Babe AND p.nameLast = Ruth
- List all players with more than 10 home runs during the 1954 season.
    - SELECT p.nameFirst p.nameLast
      FROM Players AS p, PlayerBatting as pb
      WHERE p.playerID = pb.playerID AND HR > 10

(6) Provide a plan for how you will load the database with values. – If you plan to extract/import data from on-line sources, briefly describe what are the sources (e.g. personal data, or provide URL's) and what are any format conversion issues you expect to encounter. – If you plan to input your data primarily through a web or form-based interface, briefly describe this interface and the issues involved.

Lahman Baseball Database: http://www.seanlahman.com/baseball-archive/statistics/
Retrosheet: https://www.retrosheet.org/
Baseball Reference: https://www.baseball-reference.com/

For this baseball database project we plan to extract the data from online sources. Our main source that we are looking at is the Lahman Baseball Database. The Lahman Baseball Database contains complete batting and pitching statistics from 1871 to 2019, plus fielding statistics, standings, team stats, managerial records, post-season data, and more. For our purposes this database should be more than enough to answer all of our questions. Furthermore, the information is the Lahman Database can be downloaded in SQL Lite, MySQL, and Excel so we have options in terms of how we plan on extracting the data. Some format conversion issues we might encounter include having information in our tables that we might not need. So we would potentially need to prune the data beforehand to get rid of extraneous fields. Another issue might be, if we download the data using Excel, is learning how to convert data from Excel into SQL. I believe this shouldn't be too much of a problem since conversion of data from Excel to SQL seems to be from my experience a common data science task.

(7) Very briefly describe the form/type of output or result you plan to generate or any special user interface issues (e.g. views) that you plan to implement.

We expect to generate a form/type of output where a user can select what they are looking for and insert information according to that and they will then be taken to another page where they can view the information in table form. For example lets say a user wants to see all teammates of Yogi Berra then they can select find teammates and be taken to another page where they can enter the name of the player they want to find all teammates of. From there they will be taken to another page where all of this information is displayed in table form. Some views that we plan to generate are views with a combination of information from Players table and Batting Table, Players Table and Pitching Table, and Players table and Teams table.

(8) What are the specialized/advanced topics you plan to focus on in your database design? Examples include: – security (e.g. banking) – object-oriented or distributed database design/implementation issues – advanced SQL topics (triggers, cursors, JDBC, etc.) – optimization/tuning – data mining – complex data extraction issues from online data sources – natural language interfaces – particularly advanced GUI form interface and/or report generation

- data mining (major)
    - We plan on having a focus on data mining for this project. We plan to mine our data from online sources with forms of data inputs such as Excel tables and MySQL tables that we can download from baseball statistics websites. Furthermore, we can also mine data from websites such as Baseball Reference where we can take the tabular data from these websites and input it into Excel and from there create MySQL tables from this Excel data.
- particularly advanced GUI form interface (minor)

- For this after talking with Professor Yarowsky we will be implementing some form of graphs for player statistics. Baseball statistics can be very well represented and compared in graph form. So the output of some inputs comparing multiple players can be outputted through graphs.