CIS 550 Project Description

Ryan Corkrean, Iris Gallo, Min Kim, Jeremy Kogan

October 12, 2021

1 Motivation

Baseball, more than any other major sport, places a heavy emphasis on records and statistics. From Joe DiMaggio's 56-game hit streak, to Cal Ripken Jr.'s 2,632 consecutive games played, to Tony Mullane's 343 career wild pitches (OK, maybe not that last one), these otherwise-vapid numbers provide context to the grand historical arc of America's pastime. Prior to 2015, 135 years' worth of baseball data could be stored on a 2-gigabyte flash drive. However, that season saw the installation of marker-free motion tracking equipment (TrackMan Doppler radar gave way to Hawk-Eye high-speed cameras in 2020) and the dawn of the so-called "Statcast era". Now, each game generates roughly one terabyte of data, which has informed and accelerated the development of advanced quantifications of players' talents and performances. A number of well-reputed, public-facing websites have jumped aboard baseball's big-data revolution, each presenting a subset of statistics that provide a unique snapshot of the game and its participants. As the Statcast era concludes its eighth season, two websites have cemented their products as the gold standard for baseball analysis: Baseball Savant and FanGraphs.

While these sites are extremely useful resources for the enlightened baseball fan, their true power comes from being able to survey both databases simultaneously. In this regard, it is often highly frustrating to bounce back and forth between APIs in order to unearth a particular statistic, or to perform studies that incorporate reports from both sites (especially since the two sites do not share a common player-indexing system).

As an extension of the web design concepts explored in our second homework assignment, we propose a holistic baseball API that aggregates and curates statistics from both Baseball Savant and FanGraphs, allowing fans to engage with the available data in its fullest capacity. We believe that this will facilitate more inclusive independent analysis and bring awareness to poorly-understood statistical correlations that might provide certain competitive advantages for teams and players.

2 Principal Features

The defining feature of our API will be a series of comprehensive leaderboards that amalgamate Baseball Savant's and FanGraphs respective metrics to create a streamlined, flexible, and engaging user experience. Batting and pitching statistics will be displayed in separate tables, and there will be four high-level aggregations that provide different levels of statistical specificity: overall MLB statistics, league statistics (American or National League), divisional statistics (West, Central, and East for both leagues), team statistics, and individual player statistics. Moreover, users may view results therein for individual seasons, one or more seasons (with separate records for each season), or aggregations over a range of seasons. These tables will encompass all of the information contained within our database, and will be the primary interface by which users will compare players. We also plan to implement additional complementary features that will allow users to engage with the data in different ways, including individual player pages that display all his information in a more condensed format, a strike zone heat map tool for both batters and pitchers to visualize their respective pitch-location tendencies, a spray chart for batters to show where their batted balls landed in the field of play, and a pitch arsenal graphic for pitchers to illustrate pitch type frequency, velocity, movement, spin rate, etc.

3 Ancillary Features

Given time, an additional leaderboard displaying pitch-level statistics for batters and pitchers (e.g., a table showing players' performances against four-seam fastballs in a season or seasons); however, this would necessitate a greater degree of granularity than our dataset currently accommodates, and may not be possible given the available public-facing resources. Another interesting API feature would be a scatter plot generator to allow users to analyze correlations between pairs of statistics, or a line graph generator over one or more statistics to assess temporal trends.

4 Pages

4.1 Landing/Home Page

This will be the page that the user is initially directed to upon visiting the site. It will give the user the option to display one of the leaderboards, search for a player by name, or navigate to one of the visualization tools.

4.2 Leaderboards

At the top of the leaderboards page will be a series of tabs that specify batter/pitcher leaderboards, overall/league/division/team/player leaderboards, and a series of drop-downs and check boxes that will allow the user to select data over a range of seasons and how to aggregate said data. Clicking on a player's name as it appears in a leaderboard will open their player page in a new window.

4.3 Player Pages

Individual player pages will display their yearly statistics in a more condensed format, possibly along with additional biographical information (age, current team, etc.)

4.4 Strike Zone Heatmaps

This page will display a visualization of the strike zone for a chosen batter or pitcher. Color variation can be chosen to correspond to one of pitch rate, swing rate, or contact rate.

4.5 Spray Charts

This page will display a visualization of a batter's batted ball locations within the field of play. Data point color will be used to indicate singles, doubles, triples, and home runs.

4.6 Pitch Arsenals

This page will display a pie chart of pitch type frequencies for a particular pitcher, a graph plotting pitch velocity vs. spin rate for each pitch type, and a graph plotting vertical movement vs. horizontal movement for each pitch type.

5 Relational Schema

The ER diagram for our data structure is shown in Figure 1. Observe that Batter and Pitcher are subclasses of Player (they constitute a class hierarchy), and that batters and pitchers have different sets of seasonal statistics as attributes. Since we have scraped over 100 statistics for both batters and pitchers, enumerating each subclass' attribute set would make the ER diagram utterly unreadable, so we denote the full set of Batter and Pitcher attributes with a single global attribute in the ER diagram. Some examples of Batter attributes are games played, hits, home runs, and runs batted in, while some examples of pitcher attributes are innings pitched, strikeouts, walks, and earned runs allowed. Note also that, while every player must

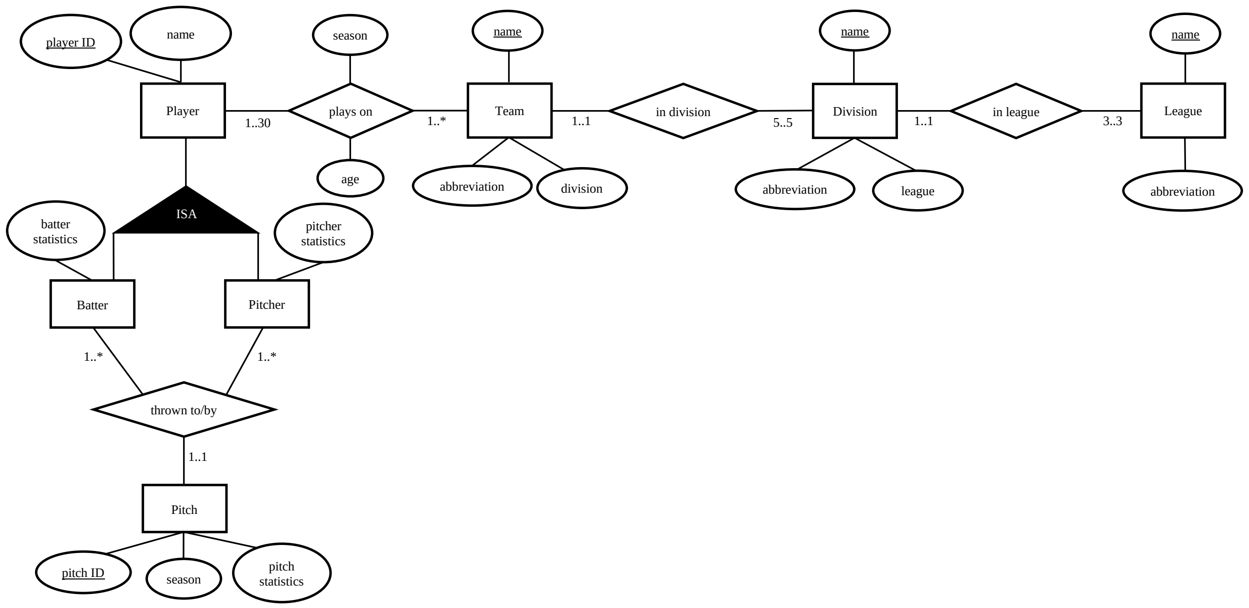

Figure 1: ER Diagram of Major League Baseball.

act as either a batter or a pitcher (that is, there is a covering constraint on Player), there is no overlap constraint in this class hierarchy, so a player may act as both a batter and a pitcher in the same season. Batter, Pitcher, and Pitch participate in a ternary relationship; every pitch in a baseball game must be thrown by exactly one pitcher to exactly one batter, a pitcher can throw one or more pitches in his career, and a batter can receive one or more pitches in his career. This dynamic is reflected in the respective participation constraints in the ternary relationship. Note that, even though there is no overlap constraint in the Player/Batter/Pitcher class hierarchy, the batter and pitcher associated with a given pitch cannot be the same player. (A single player cannot throw a pitch to himself, unless his name is Bugs Bunny!) So, there is an important fundamental constraint that cannot be captured in an ER diagram. A player may play for one or more teams during their. (In practice, this number is rarely larger than three or four, but there is no rule-imposed limit. Edwin Jackson managed to play for 14 different teams in 17 professional seasons.) A team is a part of a single division (each division having five teams), a division is a part of a league (each league having three divisions), and teams, divisions, and leagues are all identified by their bequeathed name. Players are identified by player ID (which is native to Baseball Savant), and discrete player records are identified by the identity of the player, the season in question, and the team played for. This allows for distinct records corresponding to stints with different teams in the same season, and is consistent with the many-to-many relationship depicted in the diagram. Pitches are identified solely by their pitch ID.

6 Data Definition Language

The following is a rough draft of the data definition language (DDL) that will be used to create the database.

```
CREATE DATABASE MLB;

USE MLB;

CREATE TABLE League (Name VARCHAR(50),
                     Abbreviation CHAR(2),
                     PRIMARY KEY (Name));
```

```
CREATE TABLE Division (Name VARCHAR(50),
                       Abbreviation VARCHAR(10),
                       League VARCHAR(50) NOT NULL,
                       PRIMARY KEY (Name),
                       FOREIGN KEY (League) REFERENCES League);

CREATE TABLE Team (Name VARCHAR(50),
                   Abbreviation CHAR(3),
                   Division VARCHAR(50) NOT NULL,
                   PRIMARY KEY (Name),
                   FOREIGN KEY (Division) REFERENCES Division);

CREATE TABLE Players (PlayerID INT,
                      Name VARCHAR(50) NOT NULL,
                      PRIMARY KEY (PlayerID));

CREATE TABLE Roster (PlayerID INT,
                     Team VARCHAR(50),
                     Season INT,
                     Age INT NOT NULL,
                     PRIMARY KEY (PlayerID, Team, Season),
                     FOREIGN KEY (PlayerID) REFERENCES Player,
                     FOREIGN KEY (Team) REFERENCES Team);

CREATE TABLE Batters (PlayerID INT,
                      Season INT,
                      Team VARCHAR(50),
                      Handedness CHAR(1) NOT NULL,
                      Games INT,
                      BattedBalls INT,
                      ...,
                      BattingAverage DECIMAL,
                      ExcpectedBattingAverage DECIMAL,
                      SluggingPercentage DECIMAL,
                      ExpectedSluggingPercentage DECIMAL,
                      ...,
                      WalkPercentage DECIMAL,
                      StrikeoutPercentage DECIMAL,
                      BB/K FLOAT,
                      ...,
                      wRC+ INT,
                      wOBA DECIMAL,
                      ExpectedwOBA DECIMAL,
                      ExpectedwOBACON DECIMAL,
                      ...,
                      SwingPercentage DECIMAL,
                      ChasePercentage DECIMAL,
                      SwingingStrikePercentage DECIMAL,
                      ...,
                      Barrels INT,
                      SolidContacts INT,
                      Flares INT,
                      ...,
                      LanchAngle FLOAT,
                      AverageExitVelocity FLOAT,
                      MaxExitVelocity FLOAT,
                      AverageDistance FLOAT,
                      MaxDistance FLOAT,
                      HardHit/BattedBall DECIMAL,
                      HardHit/Swing DECIMAL,
                      Barrels/BattedBall DECIMAL,
                      Barrels/PlateAppearance DECIMAL,
                      ...,
                      WinsAboveReplacement DECIMAL,
                      PRIMARY KEY (PlayerID, Season, Team),
                      FOREIGN KEY (PlayerID) REFERENCES Player,
                      FOREIGN KEY (Season) REFERENCES Roster (Season),
                      FOREIGN KEY (Team) REFERENCES Roster (Team))

CREATE TABLE Pitchers (playerID INT,
                      Season INT,
                      Team INT,
                      Handedness CHAR(1) NOT NULL,
                      Games INT,
                      BattedBalls INT,
                      ...,
                      InningsPitched DECIMAL,
                      Strikeouts INT,
                      Walks INT,
                      EarnedRunAverage DECIMAL,
                      ExpectedEarnedRunAverage DECIMAL,
                      ...,
                      StrikeoutPercentage DECIMAL,
                      WalkPercentage DECIMAL,
                      StrikeoutMinusWalkPercentage DECIMAL,
                      ...,
                      FourSeamFastballPercentage DECIMAL,
                      SinkerPercentage DECIMAL
                      CutterPercentage DECIMAL,
                      ...,
                      FastballVelocity FLOAT,
                      SinkerVelocity FLOAT,
                      CutterVelocity FLOAT,
                      ...,
                      FastballSpin FLOAT,
                      SinkerSpin FLOAT,
                      CutterSpin FLOAT,
                      ....,
                      SwingPercentage DECIMAL,
                      ChasePercentage DECIMAL,
                      SwingingStrikePercentage DECIMAL,
                      ...,
                      Barrels INT,
                      SolidContacts INT,
                      Flares INT,
                      ...,
                      LaunchAngle FLOAT,
                      AverageExitVelocity FLOAT,
                      MaxExitVelocity FLOAT,
                      AverageDistance FLOAT,
                      MaxDistance FLOAT,
                      HardHit/BattedBall DECIMAL,
                      HardHit/Swing DECIMAL,
                      Barrels/BattedBall DECIMAL,
                      Barrels/PlateAppearance DECIMAL,
                      WinsAboveReplacement DECIMAL,
                      PRIMARY KEY (PlayerID, Season, Team),
                      FOREIGN KEY (PlayerID) REFERENCES Player,
                      FOREIGN KEY (Season) REFERENCES Roster (Season),
                      FOREIGN KEY (Team) REFERENCES Roster (Team))

CREATE TABLE Pitches (PitchID INT,
                   BatterID INT NOT NULL,
                   PitcherID INT NOT NULL,
                   BatterTeam VARCHAR(50) NOT NULL,
                   PitcherTeam VARCHAR(50) NOT NULL,
                   BatterHandedness CHAR(1) NOT NULL,
                   PitcherHandedness CHAR(1) NOT NULL,
                   Season INT NOT NULL,
                      Type VARCHAR(25),
                      Velocity FLOAT,
                      SpinRate FLOAT,
                      ...,
                      Extension FLOAT,
                      PRIMARY KEY (PitchID),
                      FOREIGN KEY (BatterID) REFERENCES Batter (PlayerID),
                      FOREIGN KEY (PitcherID) REFERENCES Pitcher (PlayerID),
                      FOREIGN KEY (BatterTeam) REFERENCES Batter (Team),
                      FOREIGN KEY (PitcherTeam) REFERENCES Pitcher (Team),
                      FOREIGN KEY (BatterHandedness) REFERENCES Batter (Handedness),
                      FOREIGN KEY (PitcherHandedness) REFERENCES Pitcher (Handedness),
                      FOREIGN KEY (Season) REFERENCES Roster (Season));
```

7 Data Acquisition and Preprocessing

FanGraphs data will be acquired by generating custom player tables (using maximal granularity) through their API, which will then be saved as CSV files. The raw files will be interfaced with as pandas DataFrames, which will facilitate cleaning and aggregation. Baseball Savant's API does not allow for the same degree of customization as FanGraphs', so pitch data will be scraped using the `pybaseball` open-source Python package and stored as separate DataFrames. Players' seasonal statistics will then be rederived by aggregating this pitch data. `pybaseball` also provides player ID lookups for different APIs, which will allow us to associate a player's FanGraphs player ID with their Baseball Savant player ID and merge the two datasets appropriately. At this point, missing values will be replaced with `np.nan` (the Python equivalent of SQL's `NULL`) using the NumPy library, peripheral cleaning tasks (converting percentages to decimals, redefining team name abbreviations, etc.) will be carried out, and team; division; and league statistics will be effected by weighted means over player statistics.

8 Requisite Technologies

Our database will be stored on a MySQL RDS instance hosted on AWS, as practiced in Exercise 1, Homework 1, and Homework 2. To build our API, we will leverage React and Node.js as in Homework 2.

9 Group Responsibilities

9.1 Ryan Corkrean

Ryan will be responsible for acquiring, cleaning, and migrating the data to the RDS instance, as well as creating the API's landing page and leaderboards.

9.2 Iris Gallo

Iris will be primarily responsible for implementing the heatmap feature, and will also contribute to player page design.

9.3 Min Kim

Min will be primarily responsible for implementing the pitch arsenal visualizer, and will also contribute to player page design.

9.4 Jeremy Kogan

Jeremy will be primarily responsible for implementing the spray chart feature, and will also contribute to player page design.