Baseball Database Project
Christopher Doherty, Oliver Nabavian
CS 3200- Prof. Brown
4/29/15

Data Quality and Completeness:

Validity/Accuracy

- All player data in the batting, pitching and player tables should be 100% accurate as it was pulled from a database scraped from baseball-reference.com, a reputable source of MLB statistics.
- Data in the Team table should be accurate as well as the data was pulled directly from mlb.com and Twitter.
- Our data in the Tweet, Hashtag, Account, and User Mention tables should be accurate provided that the list of MLB player twitter handles pulled from baseball-reference is accurate. Players will often change their twitter handles if they are traded or change their number so this is cause for concern as well.

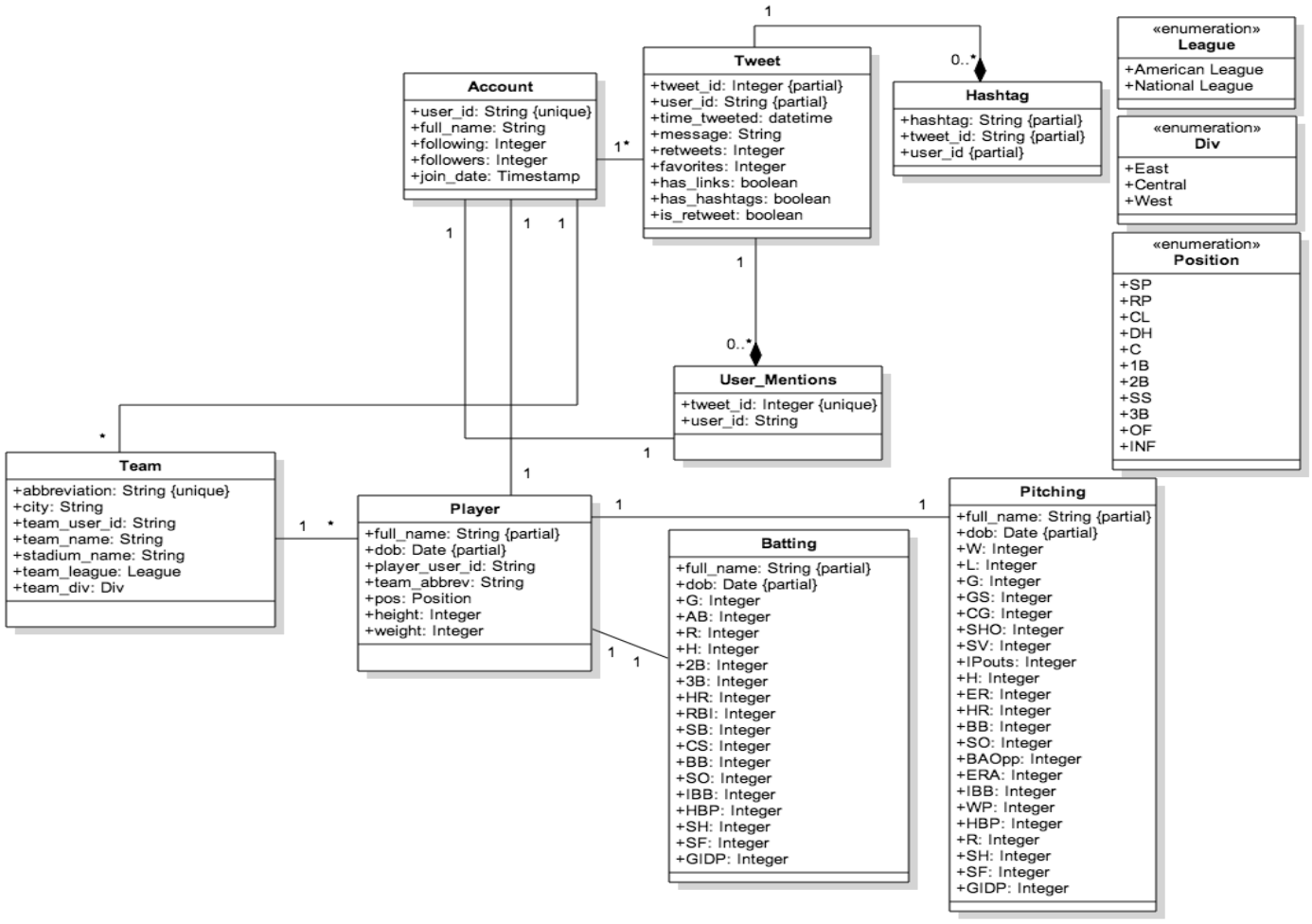
Completeness

- We have complete data for Teams, Players, player statistics, as well as complete twitter data for both team and player entities.

Consistency/Uniformity

- Our data is extremely uniform and consistent as most of it was sourced from a very clean and uniform database. The only data we have that is not consistent is the names that some players choose as their display name on twitter, as it can be anything you want and some foreign players use not English characters. This is not a problem as we can just map player name from the Player table to match with their twitter handle.

Data Schema and Relationships



Our Baseball database has 8 tables in total. Each of them consists of at least one relation. Pitching and Batting statistics are related to player names and Teams are related to players as well. Players and Teams are linked to player Twitter Accounts which are related to Tweets. Tweets are related to hashtag and user mention tables by Tweet_ids.

Data Tables and Attributes

Account: Lists every Twitter handle and some information about it

Primary Key: user_id - Each account is guaranteed to have a unique id.

```
 +-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| user_id   | varchar(25) | NO   | PRI |         |       |
| full_name | varchar(40) | NO   |     | NULL    |       |
| following | int(11)     | NO   |     | NULL    |       |
| followers | int(11)     | NO   |     | NULL    |       |
| join_date | datetime    | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
```

Player: MLB player bio information

Primary Key: (full_name, dob) - Players may have the same name but the probability of them having the same name and dob is extremely high so records are almost guaranteed to be unique.

```
+----------------+-----------------------------------------------------------+------+-----+------------+-------+
| Field          | Type                                                      | Null | Key | Default    | Extra |
+----------------+-----------------------------------------------------------+------+-----+------------+-------+
| full_name      | varchar(40)                                               | NO   | PRI | NULL       |       |
| dob            | date                                                      | NO   | PRI | 0000-00-00 |       |
| player_user_id | varchar(30)                                               | YES  |     | NULL       |       |
| team_abbrev    | varchar(3)                                                | NO   |     | NULL       |       |
| pos            |enum('1B','2B','3B','SS','C','P','LF','RF','CF','OF') | YES  |     | NULL       |
| height         | int(11)                                                   | NO   |     | NULL       |       |
| weight         | int(11)                                                   | NO   |     | NULL       |       |
+----------------+-----------------------------------------------------------+------+-----+------------+-------+
```

Team: MLB Team Bio Information

Primary Key: abbreviation - Each team in the MLB has a unique abbreviation.

```
+-----------------+-------------------------------------------+------+-----+---------+-------+
| Field           | Type                                      | Null | Key | Default | Extra |
+-----------------+-------------------------------------------+------+-----+---------+-------+
| abbreviation    | varchar(3)                                | NO   | PRI | NULL    |       |
| team_user_id    | varchar(25)                               | NO   |     | NULL    |       |
| team_name       | varchar(50)                               | YES  |     | NULL    |       |
| stadium_name    | varchar(50)                               | YES  |     | NULL    |       |
| stadium_address | varchar(255)                              | YES  |     | NULL    |       |
| team_league     | enum('National League','American League') | NO   |     | NULL    |
| team_div        | enum('East','Central','West')             | NO   |     | NULL    |       |
+-----------------+-------------------------------------------+------+-----+---------+-------+
```

Tweet: Data for each Tweet

Primary Key: (tweet_id, user_id) - Since Tweets can be retweets the user_id of the user who's timeline the tweet was taken from must be combined with the tweet id to guarantee uniqueness.

```
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| tweet_id     | varchar(100) | NO   | PRI |         |       |
| user_id      | varchar(25)  | NO   | PRI | NULL    |       |
| time_tweeted | datetime     | YES  |     | NULL    |       |
| message      | varchar(200) | NO   |     | NULL    |       |
| retweets     | int(11)      | NO   |     | NULL    |       |
| favorites    | int(11)      | NO   |     | NULL    |       |
| has_links    | tinyint(1)   | YES  |     | 0       |       |
| has_hashtags | tinyint(1)   | NO   |     | NULL    |       |
| is_retweet   | tinyint(1)   | NO   |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
```

Batting: Hitting Stats for Players.

Primary Key: (full_name, dob) - For the same reason as the player table these two columns must be combined into a composite key. This also acts as a foreign key to the *Player* table.

G: games played
AB: At-Bats
R: Runs
H: Hits
2B: Doubles
3B: Triples
HR: Homeruns
RBI: Runs Batted-In
SB: Stolen Bases
CS: Caught Stealing
BB: Walks
SO: Strikeouts
IBB: Intentional Walks
HBP: Hit By Pitch
SH: Sacrifice Bunt
SF: Sacrifice Fly
GIDP: Ground into Double Play

```
+-----------+-------------+------+-----+------------+-------+
| Field     | Type        | Null | Key | Default    | Extra |
+-----------+-------------+------+-----+------------+-------+
| G         | int(11)     | NO   |     | 0          |       |
| AB        | int(11)     | NO   |     | 0          |       |
| R         | int(11)     | NO   |     | 0          |       |
| H         | int(11)     | NO   |     | 0          |       |
| 2B        | int(11)     | NO   |     | 0          |       |
| 3B        | int(11)     | NO   |     | 0          |       |
| HR        | int(11)     | NO   |     | 0          |       |
| RBI       | int(11)     | NO   |     | 0          |       |
| SB        | int(11)     | NO   |     | 0          |       |
| CS        | int(11)     | NO   |     | 0          |       |
| BB        | int(11)     | NO   |     | 0          |       |
| SO        | int(11)     | NO   |     | 0          |       |
| IBB       | int(11)     | NO   |     | 0          |       |
| HBP       | int(11)     | NO   |     | 0          |       |
| SH        | int(11)     | NO   |     | 0          |       |
| SF        | int(11)     | NO   |     | 0          |       |
| GIDP      | int(11)     | NO   |     | 0          |       |
| full_name | varchar(40) | NO   | PRI | NULL       |       |
| dob       | date        | NO   | PRI | 0000-00-00 |       |
+-----------+-------------+------+-----+------------+-------+
```

Pitching: Statistics for Pitchers

Primary Key: (full_name, dob) - For the same reason as the player table these two columns must be combined into a composite key. This also acts as a foreign key to the *Player* table.

W: Wins
L: Losses
G: Games played
GS: Games Started
CG: Complete Games
SHO: Shutout
SV: Save
IPouts: Outs pitched (Innings Pitched * 3)
H: Hits Allowed
ER: Earned Runs
HR: Home Runs Allowed
BB: Walks
SO: Strikeouts
BAOpp: Oppenent Batting Average
ERA: Earned Run Average
IBB: Intentional Walks
WP: Wild Pitches
HBP: Hit By Pitches
SH: Sacrifice Hits
SF: Sacrifice Flys
GIDP: Ground into Double Play

```
+-----------+-------------+------+-----+------------+-------+
| Field     | Type        | Null | Key | Default    | Extra |
+-----------+-------------+------+-----+------------+-------+
| W         | int(11)     | NO   |     | 0          |       |
| L         | int(11)     | NO   |     | 0          |       |
| G         | int(11)     | NO   |     | 0          |       |
| GS        | int(11)     | NO   |     | 0          |       |
| CG        | int(11)     | NO   |     | 0          |       |
| SHO       | int(11)     | NO   |     | 0          |       |
| SV        | int(11)     | NO   |     | 0          |       |
| IPouts    | int(11)     | YES  |     | NULL       |       |
| H         | int(11)     | NO   |     | 0          |       |
| ER        | int(11)     | NO   |     | 0          |       |
| HR        | int(11)     | NO   |     | 0          |       |
| BB        | int(11)     | NO   |     | 0          |       |
| SO        | int(11)     | NO   |     | 0          |       |
| BAOpp     | double      | YES  |     | NULL       |       |
| ERA       | double      | YES  |     | NULL       |       |
| IBB       | int(11)     | YES  |     | NULL       |       |
| WP        | int(11)     | NO   |     | 0          |       |
| HBP       | int(11)     | YES  |     | NULL       |       |
| R         | int(11)     | NO   |     | 0          |       |
| SH        | int(11)     | NO   |     | 0          |       |
| SF        | int(11)     | NO   |     | 0          |       |
| GIDP      | int(11)     | YES  |     | NULL       |       |
| full_name | varchar(40) | NO   |     | NULL       |       |
| dob       | date        | NO   |     | 0000-00-00 |       |
+-----------+-------------+------+-----+------------+-------+
```

Hashtags: Table for Hashtags

Primary Key: (tweet_id, user_id, tag) - Since each tweet can contain many hashtags a composite key of the user, the id of the tweet and the contents of the hashtag must be combined to guarantee uniqueness.

```
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| tweet_id | varchar(100) | NO   | PRI |         |       |
| user_id  | varchar(15)  | NO   | PRI |         |       |
| tag      | varchar(140) | NO   | PRI | NULL    |       |
+----------+--------------+------+-----+---------+-------+
```

User_Mentions: Table showing user mentions by tweet_id

Primary Key: (tweet_id, user_id, tag) - Since each tweet can contain many user mentions, a composite of the tweet id and the user that was mentioned must be used to guarantee uniqueness.

```
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| tweet_id | varchar(100) | NO   | PRI |         |       |
| user_id  | varchar(15)  | NO   | PRI |         |       |
+----------+--------------+------+-----+---------+-------+
```

Use Cases

Use Case 1

Description: See which Players have the most followers
Actor: User
Precondition: Players must have Twitter accounts
Steps: Find all players with Twitter accounts and then find the most followed(Top 15)
Actor action: Request to see Players with Twitter accounts
System Responses: Return list of 15 players with full_names and Twitter handles
Post Condition: User will be given name and handle of most followed players
Alternate Path:
Error: User input is incorrect

```
SELECT a.full_name, a.user_id, a.followers
FROM (Account a LEFT JOIN Player p ON p.player_user_id = a.user_id)
ORDER BY a.followers DESC
LIMIT 15;
```

Use Case 2

Description: Get a Players Twitter handle with their 2014 hits
Actor: User
Precondition: Player must have a Twitter account to be included
Steps: Find all players with Twitter accounts and then find each players hits
Actor action: Request to see Players with Twitter accounts
System Responses: Return list of all players on Twitter with their 2014 hits
Post Condition: User will be given name and handle as well as hits of players
Alternate Path:
Error: User input is incorrect

```
SELECT p.full_name, p.player_user_id, b.H
FROM ((Player p INNER JOIN Batting b ON b.full_name=p.full_name AND b.dob=p.dob)
LEFT JOIN Account a ON a.user_id=p.player_user_id)
WHERE b.h > 0 AND NOT p.player_user_id="NULL"
ORDER BY b.H DESC;
```

Use Case 3

Description: Get all players who played for Mets in 2014 ordered by games played
Actor: User
Precondition: Only Includes players on the Mets
Steps: Find all players who played for the Mets in 2014 and then order them by games played

Actor action: Request to see amount of games played by each player on the Mets in 2014
System Responses: Return a list of all players who played for the Mets in 2014 ordered
from most games played to least games played
Post Condition: User will be given a list of all Mets players ordered by Games played
Alternate Path:
Error: User input is incorrect

```
SELECT p.full_name, b.G
FROM Player p
INNER JOIN Batting b
ON p.full_name=b.full_name and p.dob = b.dob
WHERE p.team_abbrev="NYM"
ORDER BY b.G DESC;
```

Use Case 4
Description: Get top 20 starting pitchers by ERA
Actor: User
Precondition: Only includes pitchers who started more than 10 games
Steps: Find all pitchers who have started more than 10 games and order them by ERA, lowest first
take the top 20 from the result
Actor action: Request to see best starting pitchers by ERA
System Responses: Return each pitchers full name, season ERA, and their team
Post Condition: User will be given a list of the top 20 pitchers, their ERA's and the team they play for.
Alternate Path:
Error: User input is incorrect

```
SELECT pi.full_name, pi.ERA, pl.team_abbrev as team
FROM Pitching pi
INNER JOIN (Player pl)
ON pi.full_name=pl.full_name and pi.dob=pl.dob
WHERE GS > 10
ORDER BY ERA
LIMIT 20;
```

Use Case 5
Description: List players, their twitter handles, and their HR's/RBI's
Actor: User
Precondition: Only includes players on Twitter
Steps: Find all players with a Twitter handle and then get their RBI's and HR's
Actor action: Request to see players with their twitter handles, HR's and RBI's
System Responses: Return each players full name, twitter handle, and their HR's/RBI's
Error: User input is incorrect

```
SELECT  p.full_name, p.player_user_id, b.HR, b.RBI
FROM Batting b
INNER JOIN Player p
ON p.full_name=b.full_name AND b.dob=p.dob
WHERE b.RBI > 0 AND NOT p.player_user_id="NULL"
ORDER BY b.RBI DESC;
```

Use Case 6
List the top 20 players by batting average with their age
Description: Return the top hitting players and their age

Actor: User
Precondition: Player must have at least 100 AB's in 2014 season
Actor action: User request for all player names as well as their hits/ab's to get avg
Steps: select player name and inner join with bating hits and at bats to get averages.
System Responses: Gets list of 20 players
Post Condition: 20 best hitters
Alternate Path: Does NOT fail

```
SELECT player.full_name
 ,timestampdiff(year, player.dob, CURRENT_DATE ()) AS age
 ,batting.h / batting.ab AS avg
FROM Player
INNER JOIN batting ON batting.full_name = player.full_name
WHERE batting.h / batting.ab >= .3
 AND batting.ab >= 100
ORDER BY avg DESC limit 20;
```

Use Case 7
Join all the 2B in the MLB and order by 2014 HR numbers most to least
Description: Return all 2B by HR
Actor: User
Precondition: Player must by primarily a 2B (most games at that position)
Actor action: User request for all player with pos as 2B as well as HR
Steps: inner join on batting stats and sorted by HR most to least
System Responses: Gets list of all 2B
Post Condition: 2B by HR numbers
Alternate Path: Does not fail

```
SELECT player.full_name
 ,player.pos
 ,batting.HR
FROM player
INNER JOIN batting ON batting.full_name = player.full_name
WHERE player.pos = '2B'
ORDER BY HR DESC;
```

Use Case 8
Join all the players names who have twitter handles with their league (national or american)
Description: Return all players on twitter who played in 2014 by first name
Actor: User
Precondition: Player must have played in 2014 and have a twitter handle
Actor action: User requests Twitter handles and team league
Steps: inner join twitter handle and player name with team league
System Responses: All players on Twitter with league
Post Condition: Players by first name
Alternate Path: Does not fail

```
SELECT player.full_name
 ,player.player_user_id
```

```
 ,team.team_league
FROM player
INNER JOIN team ON player.team_abbrev = team.abbreviation
WHERE player.player_user_id IS NOT NULL
 AND player.player_user_id <> 'null';
```

Use Case 9

Join all pitchers who had over 4 wins in 2014 and have a twitter handle
Description: Return all pitchers by wins
Actor: User
Precondition: Player must have over 4 wins in 2014 and have a twitter handle
Actor action: User requests pitching win stats and twitter handles
Steps: left join on pitching wins to Player. Get wins
System Responses: Sorted list by wins of all pitchers on twitter w/ more than 4 wins
Post Condition: Pitchers by wins
Alternate Path: Does not fail

```
SELECT player.full_name
 ,player.player_user_id
 ,pitching.w
FROM player
LEFT JOIN pitching ON player.full_name = pitching.full_name
LEFT JOIN account ON player.player_user_id = account.user_id
WHERE pitching.W > 4
 AND player.player_user_id <> 'null'
ORDER BY pitching.W DESC;
```

Use Case 10

Join players by weight and homeruns
Description: sort players by weight
Actor: User
Precondition: Must have played baseball in 2014
Actor action: User request for all baseball players weight and hr stats
Steps: left join on from players to batting stats
System Responses: List of players sorted by weight
Post Condition: heaviest to skinniest players with HR in 2014
Alternate Path: Does not fail

```
SELECT player.full_name
 ,player.weight
 ,batting.HR
FROM player
LEFT JOIN batting ON player.full_name = batting.full_name
ORDER BY player.weight DESC;
```