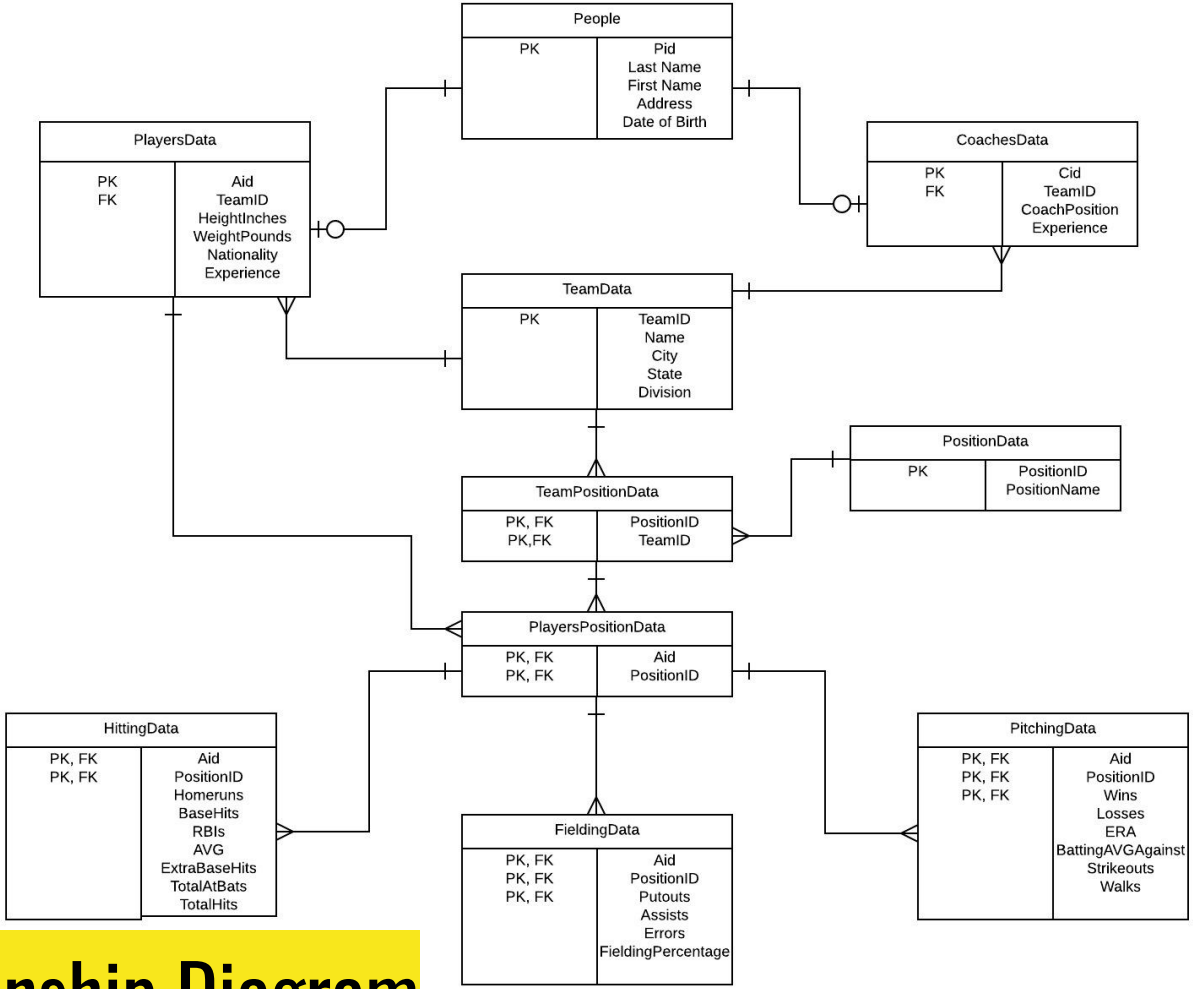
Major League Baseball: American League DB

This database was brought to you by:
Ronald Cavaliere

Table of Contents

Executive Summary

It was the spring of ‘17 when the MLB found out their steroid problem wasn’t only a problem on the field, but was a problem in their back office as well. As a result of this, their entire staff was fired and replaced. That is where I come into the picture. I was hired by director Alan Labouseur as a database designer for the MLB since their entire database got wiped out from the disgruntled, raging, steroid using workers. My job was to build a database strictly to showcase the statistics of current players in the american league. Only those who have access to this database can see it which include the coaches, players, and administration. From those who have access to it, only the American League and MLB administration teams can edit this database. This database will assist viewers for many reasons such as the All-Star selection process, and trading/signing players and coaches. In order for this database to be as easy to use as possible, I have created a bunch of different tables, views, stored procedures, and reports.



Entity-Relationship Diagram

People Table

The American League is full of many different players and coaches which is the reason for creating this people table. This has many sub-entity tables calling to it.

```
CREATE TABLE        People (
    Pid             varchar(10) not null unique,
    LastName        text not null,
    FirstName       text not null,
    Address         varchar,
    DateOfBirth     date,
    Primary key     (Pid)
);
```

FUNCTIONAL DEPENDENCIES

Pid → LastName, FirstName, Address, DateOfBirth

	pid character varying(10)	lastname text	firstname text	address character varying	dateofbirth date
1	p01	Jeter	Derek	475 Atlantic Ave	2066-07-29
2	p02	Perez	Salvador	37 Walnut Road	1972-02-15
3	p03	Hosmer	Eric	189 Cherry Lane	1983-09-26
4	p04	Altuve	Jose	28 Allstar Court	1989-04-21
5	p05	Machado	Manny	83 Braham Street	1980-01-30
6	p06	Trout	Mike	3 Jersey Road	1991-09-17
7	p07	Betts	Mookie	738 North Ave	1986-12-13
8	p08	Lindor	Fransisco	827 South Lane	1988-03-22
9	p09	Beltran	Carlos	45 West Road	1983-10-05
10	p10	Trumbo	Mark	91 East Street	1984-08-24
11	p11	Cavaliere	Ron	47 Braham Ave	1995-11-06
12	p12	Harris	Mark	39 Soccer Street	1993-08-19
13	p13	Nemergut	Paul	90 Frisbee Court	1988-03-03
14	p14	Colatosti	Ron	27 Oak Street	1991-06-12
15	p15	Graham	Zach	384 Delafield Road	1983-08-31
16	p16	Salonia	Joe	782 House Block	2052-02-17
17	p17	Sale	Chris	38 Pacific Street	1990-07-19
18	p18	Hamels	Cole	29 Indian Ave	1987-01-09
19	p19	Cano	Robinson	36 Palace Road	1985-04-18
20	p20	Girardi	Joe	19 Sunrise Ave	2049-05-11

TeamData

This table was created in order to give each team in the league a separate ID. The teamID represented in this chart acts as a foreign key to many of the other tables.

```
CREATE TABLE        TeamData (
        TeamID          varchar(10) not null,
        TeamName            text not null,
        City            text not null,
        State           text not null,
        Division        text not null,
        Primary key     (TeamID)
);
```

FUNCTIONAL DEPENDENCIES

TeamID →TeamName, City, State, Division

	teamid character varying(10)	teamname text	city text	state text	division text
1	c01	New York Yankees	Bronx	New York	American League East
2	c02	Boston Red Sox	Boston	Massachusetts	American League East
3	c03	Cleveland Indians	Cleveland	Ohio	American League Central
4	c04	Toronto Blue Jays	Toronto	Canada	American League East
5	c05	Baltimore Orioles	Baltimore	Maryland	American League EaSt
6	c06	Texas Rangers	Arlington	Texas	American League West
7	c07	Kansas City Royals	Kansas City	Missouri	American League Central
8	c08	Los Angeles Angels	Anaheim	California	American League West
9	c09	Chicago White Sox	Chicago	Illinois	American League Central
10	c10	Oakland Athletics	Oakland	California	American League West
11	c11	Houston Astros	Houston	Texas	American League West
12	c12	Seattle Mariners	Seattle	Washington	American League West
13	c13	Minnesota Twins	Minneapolis	Minnesota	American League Central
14	c14	Tampa Bay Rays	Tampa	Florida	American League East
15	c15	Detroit Tigers	Detroit	Michigan	American League Central

PlayersData

This table was created in order to give each player in the league a separate ID and to assign them to the team they play for. This table refers back to the people table and teamData.

```
CREATE TABLE                PlayersData (
  Aid                 varchar(10) not null references People (Pid),
  TeamID              varchar(10) not null references TeamData (TeamID),
  HeightInches        integer not null,
  WeightPounds        integer not null,
  Nationality         text not null,
  Experience          text not null,
 primary key (Aid)
);
```

FUNCTIONAL DEPENDENCIES

Aid →TeamID, HeightInches, WeightPounds, Nationality, Experience

	aid character varying(10)	teamid character varying(10)	heightinches integer	weightpounds integer	nationality text	experience text
1	p01	c01	71	175	American	15 Years
2	p02	c07	74	215	Dominican Republican	8 Years
3	p03	c06	72	187	American	9 Years
4	p04	c09	77	192	Mexican	5 Years
5	p05	c12	73	188	Korean	7 Years
6	p06	c05	75	168	Brazilian	10 Years
7	p07	c08	72	192	American	6 Years
8	p08	c12	72	183	American	3 Years
9	p09	c10	73	177	Cuban	1 Years
10	p10	c03	68	1215	British	5 Years
11	p11	c09	71	199	Spanish	10 Years
12	p13	c11	70	180	American	2 Years
13	p15	c06	75	210	Norwegian	8 Years
14	p17	c14	73	192	Irish	10 Years
15	p18	c05	77	195	Italian	9 Years
16	p19	c06	69	168	Canadian	8 Years

CoachesData

This table was created in order to give each coach in the league a separate ID. It refers back to the people table and teamData.

```
CREATE TABLE CoachesData (
  Cid                   varchar(10) not null references People (Pid),
  TeamID                varchar(10) not null references TeamData (TeamID),
  CoachPosition         text,
  CoachingExperience    text,
  primary key (Cid)
);
```

	cid character varying(10)	teamid character varying(10)	coachposition text	coachingexperience text
1	p12	c01	Head Coach	25 Years
2	p14	c07	Head Coach	5 Years
3	p16	c13	Head Coach	17 Years
4	p20	c04	Head Coach	9 Years

FUNCTIONAL DEPENDENCIES

Cid →TeamID, CoachPosition, CoachingExperience

PositionData

This table was created in order to identify and label each position. It acts as a foreign key to many tables.

```
CREATE TABLE PositionData (
  PositionID            text unique not null,
  PositionName          text not null,
  primary key (PositionID)
);
```

FUNCTIONAL DEPENDENCIES

PositionID → PositionName

	positionid text	positionname text
1	H	HittingData
2	P	PitchingData
3	P1	FieldingData
4	C	FieldingData
5	1B	FieldingData
6	2B	FieldingData
7	3B	FieldingData
8	SS	FieldingData
9	LF	FieldingData
10	CF	FieldingData
11	RF	FieldingData

TeamPositionData

This table was created in order to identify and label each position for each team in the league.

```
CREATE TABLE TeamPositionData (
  TeamID          varchar(10) not null references TeamData (TeamID),
  PositionID      text not null references PositionData (PositionID),
 primary key (TeamID, PositionID)
);
```

FUNCTIONAL DEPENDENCIES

TeamID, PositionID →

	teamid character varying(10)	positionid text
1	c01	H
2	c01	P
3	c01	C
4	c01	1B
5	c01	2B
6	c01	3B
7	c01	SS
8	c01	LF
9	c01	CF
10	c01	RF

PlayersPositionData

This table was created to give each player in the league a specific position. Since each hitter is also a fielder I had duplicate each Pid. Also, since pitchers do not hit, all fielders are not hitters.

```
CREATE TABLE PlayersPositionData (
  Aid              varchar(10) not null references PlayersData (Aid),
  PositionID       text not null references PositionData (PositionID),
  primary key (Aid, PositionID)
);
```

FUNCTIONAL DEPENDENCIES

Aid, PositionID →

	aid character varying(10)	positionid text
1	p01	C
2	p01	H
3	p02	1B
4	p02	H
5	p03	2B
6	p03	H
7	p04	3B
8	p04	H
9	p05	SS
10	p05	H
11	p06	LF
12	p06	H
13	p07	CF
14	p07	H
15	p08	RF
16	p08	H
17	p09	LF
18	p09	H

HittingData

This table was created in order to show the hitting stats for each player. It references back to PlayersData and PositionData in order to find hitters.

```
CREATE TABLE HittingData (
  Aid              varchar(10) not null references PlayersData (Aid),
  PositionID    text not null references PositionData (PositionID) check (PositionID = 'H'),
  Homeruns            integer,
  BaseHits      integer,
  RBIs                  integer,
  AVG                  varchar(10),
  ExtraBaseHits     integer,
  TotalAtBats       integer,
  TotalHits         integer,
 primary key (Aid, PositionID)
);
```

	aid character varying(10)	positionid text	homeruns integer	basehits integer	rbis integer	avg character varying(15)	extrabasehits integer	totalatbats integer	totalhits integer
1	p01	H	50	100	115	.350	20	500	300
2	p02	H	22	85	105	.400	13	450	250
3	p03	H	15	90	86	.300	20	400	180
4	p04	H	9	99	75	.250	15	427	150
5	p05	H	2	205	115	.405	23	475	265
6	p06	H	38	79	110	.292	29	432	203
7	p07	H	13	95	88	.305	35	435	186
8	p08	H	21	69	27	.200	5	315	87
9	p09	H	2	55	43	.300	10	300	99
10	p10	H	45	125	119	.315	33	445	204
11	p11	H	7	102	90	.290	9	379	147
12	p19	H	18	64	172	.333	6	385	125

FUNCTIONAL DEPENDENCIES

Aid, PositionID → Homeruns, BaseHits, RBIs, AVG,
ExtraBaseHits, TotalAtBats, TotalHits

FieldingData

This table was created in order to show the fielding stats for each player. It references PlayersData and PositionData in order to find fielders.

```
CREATE TABLE FieldingData (
  Aid                  varchar(10) not null references PlayersData (Aid),
  PositionID           text not null references PositionData (PositionID)
                           check (PositionID = 'P' or  PositionID = 'C' or PositionID = '1B' or
                           PositionID = '2B' or PositionID = '3B' or PositionID = 'SS' or PositionID =
                           'LF' or PositionID = 'CF' or PositionID = 'RF'),
  Putouts              integer,
  Assists              integer,
  Errors               integer,
  FieldingPercentage   varchar(10),
 primary key (Aid, PositionID)
);
```

FUNCTIONAL DEPENDENCIES

Aid, PositionID → Putouts, Assists, Errors, FieldingPercentage

	aid character varying(10)	positionid text	putouts integer	assists integer	errors integer	fieldingpercentage character varying(10)
1	p01	C	50	45	5	.975
2	p02	1B	75	78	5	.975
3	p03	2B	63	42	0	1.000
4	p04	3B	77	39	3	.970
5	p05	SS	95	74	4	.790
6	p06	LF	70	45	21	.735
7	p07	CF	63	80	3	.835
8	p08	RF	129	32	1	.999
9	p09	LF	100	93	9	.900
10	p10	CF	93	23	2	.969
11	p11	RF	74	63	7	.889
12	p13	P	43	59	19	.763
13	p15	P	78	35	8	.910
14	p17	P	94	68	1	.995
15	p18	P	53	39	15	.805
16	p19	C	22	95	10	.855

PitchingData

This table was created in order to show the pitching stats for each player. It references back to PlayersData and PositionData in order to find pitchers.

```
CREATE TABLE PitchingData (
  Aid                    varchar(10) not null references PlayersData (Aid),
  PositionID             text not null references PositionData (PositionID)
                             check (PositionID = 'P'),
  Wins                   integer,
  Losses                 integer,
  ERA                    varchar(10),
  BattingAVGAgainst      varchar(10),
  Strikeouts             integer,
  Walks                  integer,
 primary key (Aid, PositionID)
);
```

	aid character varying(10)	positionid text	wins integer	losses integer	era character varying(15)	battingavgagainst character varying(15)	strikeouts integer	walks integer
1	p13	P	10	2	2.15	.195	101	15
2	p15	P	8	8	3.25	.210	64	9
3	p17	P	5	9	3.60	.230	59	12
4	p18	P	7	3	3.04	.215	62	7

FUNCTIONAL DEPENDENCIES

Aid, PositionID → Wins, Losses, ERA, BattingAVGAgainst, Strikeouts, Walks

Hitters View

This view was created in order to have an easy way of finding each hitter. Depending on what a coach is looking for, this can be useful during the trading process.

```
CREATE VIEW Hitters AS
select distinct p.pid, p.lastName, p.firstName, pod.positionID, td.TeamName
from PlayersData pd, playerpositiondata ppd, People p, PositionData pod, TeamPositionData tp,
TeamData td
where pod.positionId = tp.positionID
        and tp.positionID = ppd.positionID
        and ppd.aid = pd.aid
        and pd.aid = p.pid
        and td.teamID = pd.teamID
        and pod.positionName = 'HittingData'
order by pid ASC;

Select *
from Hitters;
```

	pid character varying(10)	lastname text	firstname text	positionid text	teamname text
1	p01	Jeter	Derek	H	New York Yankees
2	p02	Perez	Salvador	H	Kansas City Royals
3	p03	Hosmer	Eric	H	Texas Rangers
4	p04	Altuve	Jose	H	Chicago White Sox
5	p05	Machado	Manny	H	Seattle Mariners
6	p06	Trout	Mike	H	Baltimore Orioles
7	p07	Betts	Mookie	H	Los Angeles Angels
8	p08	Lindor	Fransisco	H	Seattle Mariners
9	p09	Beltran	Carlos	H	Oakland Athletics
10	p10	Trumbo	Mark	H	Cleveland Indians
11	p11	Cavaliere	Ron	H	Chicago White Sox
12	p19	Cano	Robinson	H	Texas Rangers

Fielders View

This view was created in order to have an easy way of finding each fielder. It displays the position he plays as well as the team he plays for. Depending on what a coach is looking for, this can be useful during the trading process.

```
CREATE VIEW Fielders AS
select distinct p.pid, p.lastName, p.firstName, pod.positionID, td.TeamName
from PlayersData pd, playerpositiondata ppd, People p, PositionData pod, TeamPositionData tp, TeamData td
where pod.positionId = tp.positionID
        and tp.positionID = ppd.positionID
        and ppd.aid = pd.aid
        and pd.aid = p.pid
        and td.TeamID = pd.TeamID
        and (pod.positionName = 'FieldingData'
        OR pod.positionName = 'PitchingData')
order by pid ASC;

Select *
from Fielders;
```

	pid character varying(10)	lastname text	firstname text	positionid text	teamname text
1	p01	Jeter	Derek	C	New York Yankees
2	p02	Perez	Salvador	1B	Kansas City Royals
3	p03	Hosmer	Eric	2B	Texas Rangers
4	p04	Altuve	Jose	3B	Chicago White Sox
5	p05	Machado	Manny	SS	Seattle Mariners
6	p06	Trout	Mike	LF	Baltimore Orioles
7	p07	Betts	Mookie	CF	Los Angeles Angels
8	p08	Lindor	Fransisco	RF	Seattle Mariners
9	p09	Beltran	Carlos	LF	Oakland Athletics
10	p10	Trumbo	Mark	CF	Cleveland Indians
11	p11	Cavaliere	Ron	RF	Chicago White Sox
12	p13	Nemergut	Paul	P	Houston Astros
13	p15	Graham	Zach	P	Texas Rangers
14	p17	Sale	Chris	P	Tampa Bay Rays
15	p18	Hamels	Cole	P	Baltimore Orioles
16	p19	Cano	Robinson	C	Texas Rangers

Pitchers View

This view was created in order to have an easy way of finding each pitcher. Depending on what a coach is looking for, this can be useful during the trading process.

```
CREATE VIEW Pitchers AS
select distinct p.pid, p.lastName, p.firstName, pod.positionID, td.TeamName
from PlayersData pd, playerpositiondata ppd, People p, PositionData pod,
TeamPositionData tp, TeamData td
where pod.positionId = tp.positionID
        and tp.positionID = ppd.positionID
        and ppd.aid = pd.aid
        and pd.aid = p.pid
        and td.TeamID = pd.TeamID
        and pod.positionName = 'PitchingData'
order by pid ASC;

Select *
from Pitchers;
commit;
```

	pid character varying(10)	lastname text	firstname text	positionid text	teamname text
1	p13	Nemergut	Paul	P	Houston Astros
2	p15	Graham	Zach	P	Texas Rangers
3	p17	Sale	Chris	P	Tampa Bay Rays
4	p18	Hamels	Cole	P	Baltimore Orioles

Reports

1) Returning all players who have more than 5 years of experience in the league

```
select p.*, td.TeamName, pd.Experience
from people p, PlayersData pd, TeamData td
where pid = aid
   and td.TeamID = pd.TeamID
   and pd.Experience  >  5;
```

	pid character varying(10)	lastname text	firstname text	address character varying	dateofbirth date	teamname text	experience integer
1	p01	Jeter	Derek	475 Atlantic Ave	2066-07-29	New York Yankees	15
2	p02	Perez	Salvador	37 Walnut Road	1972-02-15	Kansas City Royals	8
3	p03	Hosmer	Eric	189 Cherry Lane	1983-09-26	Texas Rangers	9
4	p06	Trout	Mike	3 Jersey Road	1991-09-17	Baltimore Orioles	10
5	p11	Cavaliere	Ron	47 Braham Ave	1995-11-06	Chicago White Sox	10
6	p15	Graham	Zach	384 Delafield Road	1983-08-31	Texas Rangers	8
7	p17	Sale	Chris	38 Pacific Street	1990-07-19	Tampa Bay Rays	10
8	p18	Hamels	Cole	29 Indian Ave	1987-01-09	Baltimore Orioles	9
9	p19	Cano	Robinson	36 Palace Road	1985-04-18	Texas Rangers	8

This report was created in order to have an easy way of finding each player who has more than 5 years of experience. Finding a veteran can be crucial for some teams. Depending on what a viewer is looking for, this can be useful during the trading and All-Star Selection process.

Reports

2) Returning all players who play for the most successful franchises in the American League.

```
select distinct p.pid, p.lastName, p.firstName, td.TeamName, td.Division, pd.heightInches,
pd.weightPounds
from PlayersData pd, TeamData td, People p
where p.pid = pd.aid
    and td.TeamID = pd.TeamID
    and td.TeamName IN (select distinct td.TeamName
                        from TeamData td
                        where  td.TeamName IN ('New York Yankees', 'Oakland Athletics', 'Boston
Red Sox', 'Toronto BLue Jays', 'Baltimore Orioles')
                    )
Order by td.TeamName;
```

	pid character varying(10)	lastname text	firstname text	teamname text	division text	heightinches integer	weightpounds integer
1	p06	Trout	Mike	Baltimore Orioles	American League East	75	168
2	p18	Hamels	Cole	Baltimore Orioles	American League East	77	195
3	p01	Jeter	Derek	New York Yankees	American League East	71	175
4	p09	Beltran	Carlos	Oakland Athletics	American League West	73	177

Hitting Stored Procedure

```
create or replace function HittingData(text, text,REFCURSOR) returns refcursor as $$
declare
  PlayerLastName          text      := $1;
  PlayerFirstName         text      := $2;
  answerset           REFCURSOR  := $3;
begin
  open answerset for
    select distinct p.lastname,p.firstname, td.TeamName, hd.*
    from HittingData hd, people p, TeamData td, PlayersData pd
    where hd.aid = p.pid
        and p.lastName LIKE LastName
        and p.firstName LIKE FirstName
        and hd.aid = pd.aid
        and pd.aid = p.pid
        and pd.TeamID = td.TeamID
     order by Homeruns Desc;
   return answerset;
end;
$$
language plpgsql;

select HittingData('L%', 'J%','answers');
fetch all from answers;
```

This stored procedure was created in order to have an easy way of finding the stats of each hitter. Finding a solid hitter can be very tough sometimes. Depending on what a viewer is looking for, this can be useful during the trading and All-Star Selection process.

	lastname text	firstname text	teamname text	aid character varying(10)	positionid text	homeruns integer	basehits integer	rbis integer	avg character varying(15)	extrabasehits integer	totalatbats integer	totalhits integer
1	Jeter	Derek	New York Yankees	p01	H	50	100	115	.350	20	500	300
2	Trumbo	Mark	Cleveland Indians	p10	H	45	125	119	.315	33	445	204
3	Trout	Mike	Baltimore Orioles	p06	H	38	79	110	.292	29	432	203
4	Perez	Salvador	Kansas City Royals	p02	H	22	85	105	.400	13	450	250
5	Lindor	Fransisco	Seattle Mariners	p08	H	21	69	27	.200	5	315	87
6	Cano	Robinson	Texas Rangers	p19	H	18	64	172	.333	6	385	125
7	Hosmer	Eric	Texas Rangers	p03	H	15	90	86	.300	20	400	180
8	Betts	Mookie	Los Angeles Angels	p07	H	13	95	88	.305	35	435	186
9	Altuve	Jose	Chicago White Sox	p04	H	9	99	75	.250	15	427	150
10	Cavaliere	Ron	Chicago White Sox	p11	H	7	102	90	.290	9	379	147
11	Beltran	Carlos	Oakland Athletics	p09	H	2	55	43	.300	10	300	99
12	Machado	Manny	Seattle Mariners	p05	H	2	205	115	.405	23	475	265

Position Stored Procedure

```
create or replace function Get_Position_Player(text,REFCURSOR) returns refcursor as $$
declare
   Position            text      :=  $1;
   answerset                  REFCURSOR   := $2;
begin
   open answerset for
      select distinct fd.aid, p.lastName, p.firstName, fd.positionID, td.TeamName
      from FieldingData fd, People p, TeamData td, PlayersData pd
      where fd.aid = pd.aid
         and pd.aid = p.pid
         and td.teamID = pd.teamID
         and fd.positionID = Position
       order by aid ASC;
    return answerset;
end;
$$
language plpgsql;

select Get_Position_Player('C', 'answers2');
fetch all from answers2;
```

	aid character varying(10)	lastname text	firstname text	positionid text	teamname text
1	p01	Jeter	Derek	C	New York Yankees
2	p19	Cano	Robinson	C	Texas Rangers

This stored procedure was created in order to have an easy way of finding players by their position. If a team is looking for a fill spot for a particular position this would be great to use. If injuries are a problem for your team and you need to pick somebody up quick this would be a good tool. It can also be useful during the trading and All-Star Selection process.

Team Stored Procedure

```
create or replace function Get_Player_by_Team(text,REFCURSOR) returns refcursor as $$
declare
   TeamNameInt          text       :=  $1;
   answerset          REFCURSOR  :=  $2;
begin
   open answerset for
      select distinct pd.aid, p.lastName, p.firstName
      from People p, TeamData td, PlayersData pd
      where pd.teamID = td.teamID
         and p.pid = pd.aid
        and td.teamName Like TeamNameInt
      order by aid ASC;
   return answerset;
end;
$$
language plpgsql;

select Get_Player_by_Team('Baltimore Orioles', 'answers3');
fetch all from answers3;
```

This final stored procedure was created in order to have an easy way of finding players of each team. Finding a solid player can be very tough sometimes. Depending on what a viewer is looking for, this can be useful during the trading and All-Star Selection process.

	aid character varying(10)	lastname text	firstname text
1	p06	Trout	Mike
2	p18	Hamels	Cole

Roles

MLB Back Office

```
create role MLB_Back_Office;
grant select, insert, update
on all tables in schema public
to MLB_Back_Office;
```

American League Administration

```
create role American_League_Administration;
grant select, insert, update
on all tables in schema public
to American_League_Administration;
```

MLB Players

```
create role MLB_Players;
grant select
on all tables in schema public
to MLB_Players;
```

MLB Coaches

```
create role MLB_Coaches;
grant select
on all tables in schema public
to MLB_Coaches;
```

I created these for roles in order to give editting access to both the MLB Back Office and the American League Administration. As you can see, players and coaches don't have the ability to edit this database but can view it whenever they would like to. These securities are essential when creating a efficient database.

Implementation

While creating this database I used PostgreSQL. I created all of the coding first and then tested all of it at once in the end. I would later find out that what I did was a horrible idea. Nothing worked because of silly syntax errors which ended up taking forever to find. Then I realized I created some of the stat charts with a vertical format so I had to change those to look better. I changed it to horizontal which ended up working out. It only displayed each player's ID once with their stats going across the board. One last issue was telling the database that a pitcher was a fielder but not a hitter. It wasn't picking up the position "P" in fielding data which I had to fix. Overall, this was a great experience and I learned so much from it. It required a lot of time and effort but was worth it in the end.

Known Problems/ Future Enhancements

- The biggest problem is the fact that coaches and players would not be able to filter players by a specific stat. If a viewer wanted to look up players with the most home runs in the league the rest of their stats would still be displayed.
- All of these stats are fictional and randomly inputted. One thing you might notice is that Derek Jeter is a catcher. Obviously he wasn't a catcher but it ended up that way by chance.
- All of the national league teams will need to be added
- The Designated Hitter was a position would also need to be added
- I would like to eventually add a section for umpires and their ratings
- Create a way to automatically delete players who retire
- Injury table can definitely be added since players get hurt every year