# NCAA Football Database

By: Nick Comito

## Table of Contents

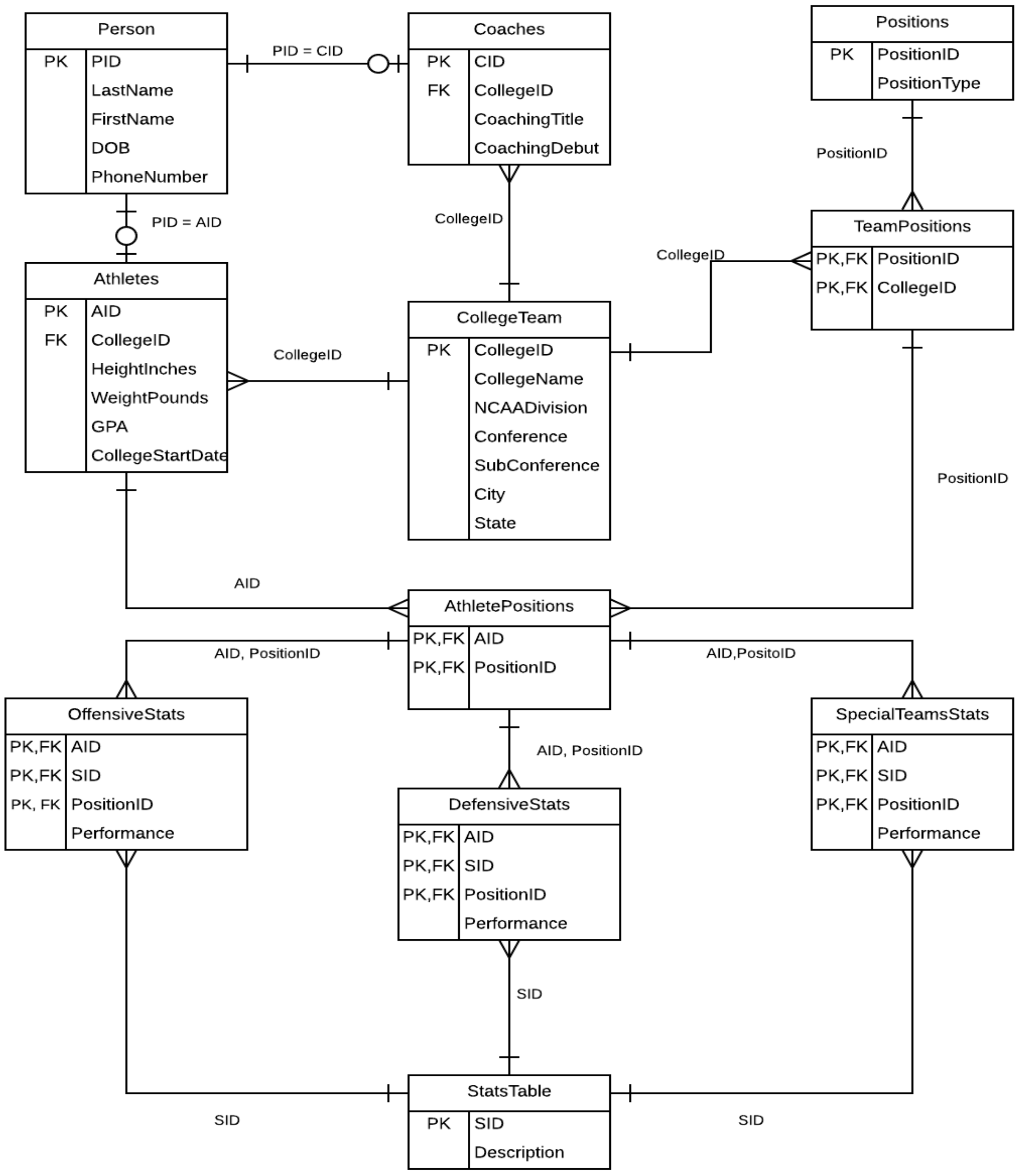
# Executive Summary

The National Colligate Athletic Association (NCAA) is one of the largest sports organizations in the world with over 1,100 college institutions and more than 480,000 student athletes. The NCAA is broken down into three divisions with Division I being the most completive followed by II and III respectively. Out of the all of the sports the NCAA has, football makes up the largest percentage of athletes compared to any other sport. In Division I alone, there are over 250 teams that on average carry around 110 players. It is important for the NCAA to have a strong relational database to track each of these teams and players for their own organizational purpose but also for teams in the National Football Team to scout and recruit potential NFL players.

The purpose of NCAA Football Database I have created is to provide NFL organizations with an effective and efficient database to track schools, players, coaches, and athlete stats from colleges and universities around the country at every division level. The objective of this database is to be user friendly in order for those with access to the database to be able to easily find the data they are looking for. In order to do so, views and store procedures have been created as well as sample reports that may be useful for the database users. Along with NFL teams, other users include college teams (coaches and administration) and the NCAA Football staff and administration. The NCAA is the only user of the database that is allowed to insert or update the database.

As one reads through this database proposal they will come across an Entity Relationship Diagram to show how the database functions and how each table relates to one another. Create statements, functional dependencies and sample data for each table are presented followed by views, stored procedures and sample reports. The final part of the proposal includes implementation comments as well as known issues and future enhancements to the database that could improve its quality.

# E/R Diagram



Person
PK PID
LastName
FirstName
DOB
PhoneNumber
PID = CID
Coaches
PK CID
FK CollegeID
CoachingTitle
CoachingDebut
Positions
PK PositionID
PositionType
PositionID
PID = AID
CollegeID
CollegeID
TeamPositions
PK,FK PositionID
PK,FK CollegeID
Athletes
PK AID
FK CollegeID
HeightInches
WeightPounds
GPA
CollegeStartDate
CollegeID
CollegeTeam
PK CollegeID
CollegeName
NCAADivision
Conference
SubConference
City
State
PositionID
AID
AthletePositions
PK,FK AID
PK,FK PositionID
AID, PositionID
AID,PositoID
OffensiveStats
PK,FK AID
PK,FK SID
PK, FK PositionID
Performance
AID, PositionID
DefensiveStats
PK,FK AID
PK,FK SID
PK,FK PositionID
Performance
SpecialTeamsStats
PK,FK AID
PK,FK SID
PK,FK PositionID
Performance
SID
SID
StatsTable
PK SID
Description
SID

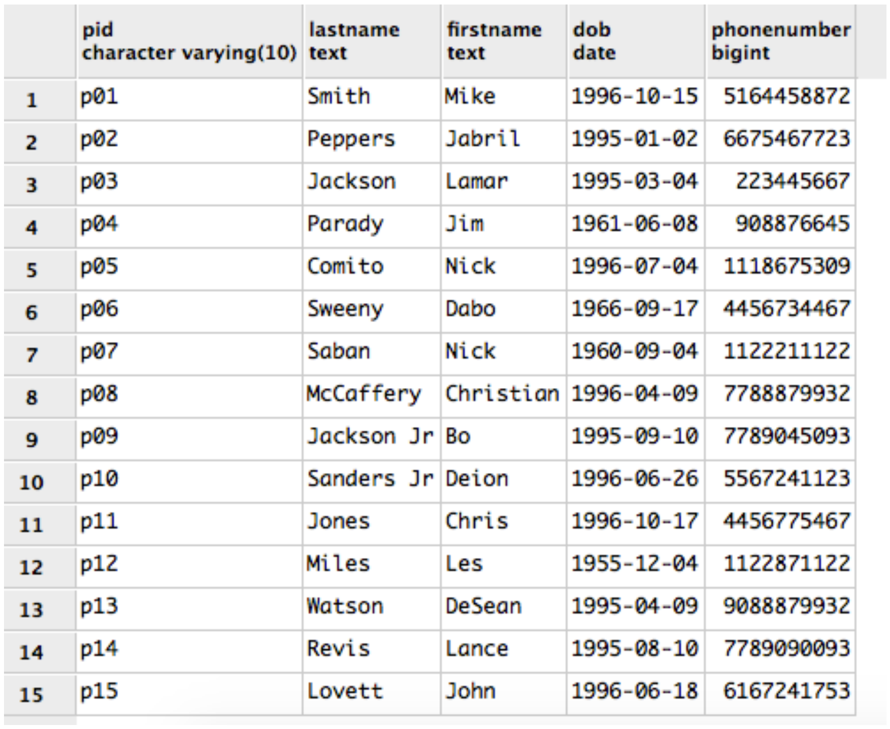
## Person Table:

Since college football teams are broken down into players and coaches, an overall person table must be created with sub-entity tables stemming from it. There is also the chance that a player could also be a coach for a team.

```
CREATE TABLE Person (
     Pid                    varchar(10) not null unique,
     LastName           text,
     FirstName          text,
     DOB                 date,
     PhoneNumber    BIGINT,
    primary key(Pid)
);
```

### Functional Dependencies:

<u>Pid</u> → LastName, FirstName, DOB, PhoneNumber



| | pid character varying(10) | lastname text | firstname text | dob date | phonenumber bigint |
|---|---|---|---|---|---|
| 1 | p01 | Smith | Mike | 1996-10-15 | 5164458872 |
| 2 | p02 | Peppers | Jabril | 1995-01-02 | 6675467723 |
| 3 | p03 | Jackson | Lamar | 1995-03-04 | 223445667 |
| 4 | p04 | Parady | Jim | 1961-06-08 | 908876645 |
| 5 | p05 | Comito | Nick | 1996-07-04 | 1118675309 |
| 6 | p06 | Sweeny | Dabo | 1966-09-17 | 4456734467 |
| 7 | p07 | Saban | Nick | 1960-09-04 | 1122211122 |
| 8 | p08 | McCaffery | Christian | 1996-04-09 | 7788879932 |
| 9 | p09 | Jackson Jr | Bo | 1995-09-10 | 7789045093 |
| 10 | p10 | Sanders Jr | Deion | 1996-06-26 | 5567241123 |
| 11 | p11 | Jones | Chris | 1996-10-17 | 4456775467 |
| 12 | p12 | Miles | Les | 1955-12-04 | 1122871122 |
| 13 | p13 | Watson | DeSean | 1995-04-09 | 9088879932 |
| 14 | p14 | Revis | Lance | 1995-08-10 | 7789090093 |
| 15 | p15 | Lovett | John | 1996-06-18 | 6167241753 |

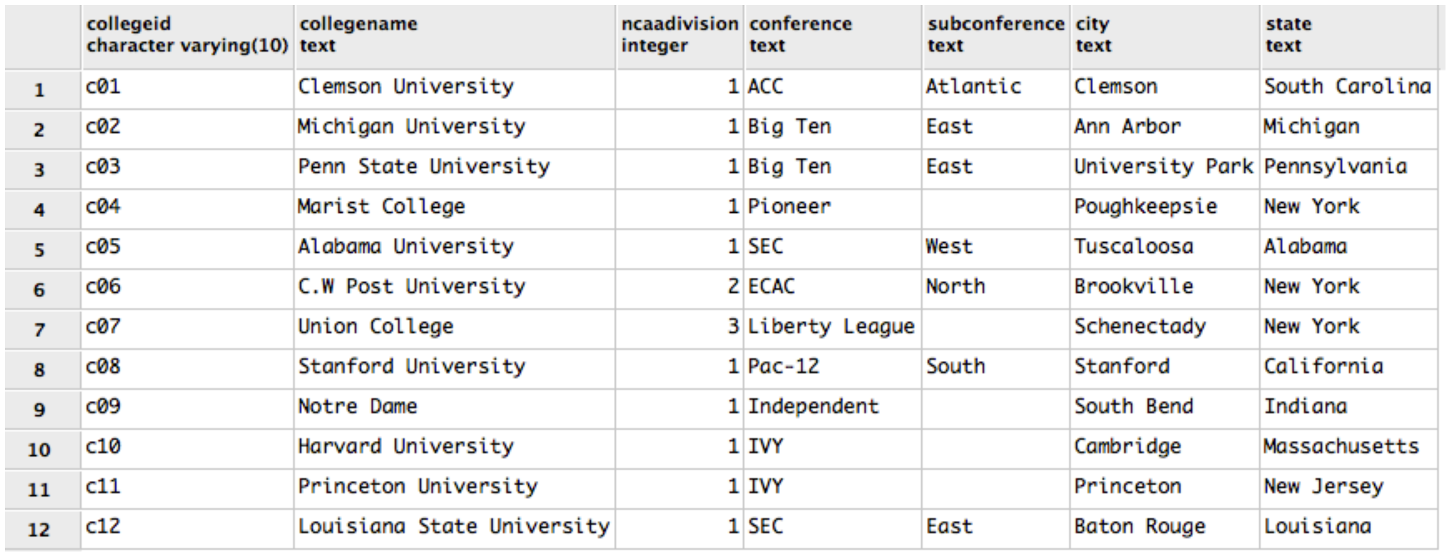
## CollegeTeams Table:

This table is a strong entity that acts as a foreign key to many tables. There is a check constraint on NCAADivision since a college division can only be 1, 2, or 3

```
CREATE TABLE CollegeTeams (
        CollegeID           varchar(10) not null,
        CollegeName         text not null,
        NCAADivision      integer not null check (NCAADivision >= 1 and NCAADivision <= 3),
        Conference          text not null,
        SubConference       text,
        City                text not null,
        State               text not null,
        primary key(CollegeID)
);
```

### Functional Dependencies:

<u>CollegeID</u> → CollegeName, NCAADivision, Conference, SubConference, City, State



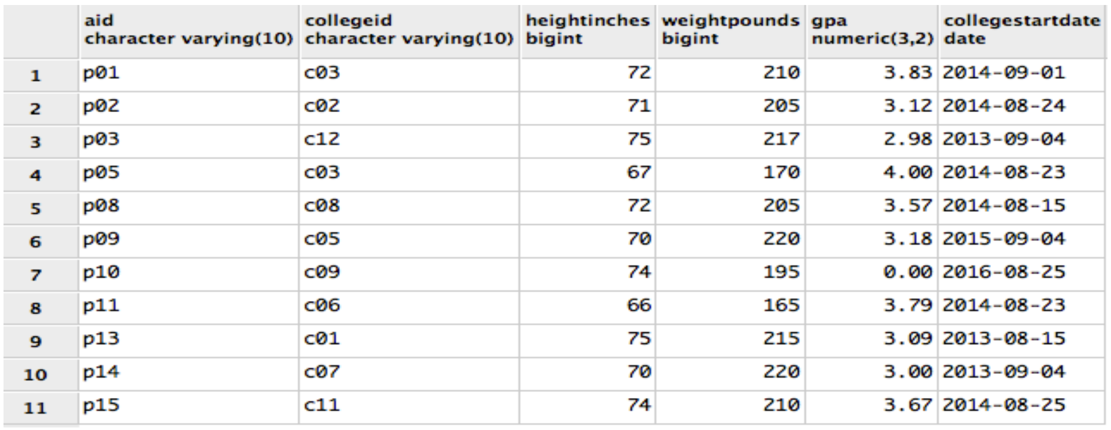
| | collegeid character varying(10) | collegename text | ncaadivision integer | conference text | subconference text | city text | state text |
|---|---|---|---|---|---|---|---|
| 1 | c01 | Clemson University | 1 | ACC | Atlantic | Clemson | South Carolina |
| 2 | c02 | Michigan University | 1 | Big Ten | East | Ann Arbor | Michigan |
| 3 | c03 | Penn State University | 1 | Big Ten | East | University Park | Pennsylvania |
| 4 | c04 | Marist College | 1 | Pioneer | | Poughkeepsie | New York |
| 5 | c05 | Alabama University | 1 | SEC | West | Tuscaloosa | Alabama |
| 6 | c06 | C.W Post University | 2 | ECAC | North | Brookville | New York |
| 7 | c07 | Union College | 3 | Liberty League | | Schenectady | New York |
| 8 | c08 | Stanford University | 1 | Pac-12 | South | Stanford | California |
| 9 | c09 | Notre Dame | 1 | Independent | | South Bend | Indiana |
| 10 | c10 | Harvard University | 1 | IVY | | Cambridge | Massachusetts |
| 11 | c11 | Princeton University | 1 | IVY | | Princeton | New Jersey |
| 12 | c12 | Louisiana State University | 1 | SEC | East | Baton Rouge | Louisiana |

## Athletes Table:

```
CREATE TABLE Athletes (
    Aid                 varchar(10) not null references Person(Pid),
    CollegeID           varchar(10) not null references CollegeTeams(CollegeID),
    HeightInches        BIGINT,
    WeightPounds        BIGINT,
    GPA                 decimal(3,2) check (GPA >= 0.00 and GPA <= 4.00),
    CollegeStartDate    date not null,
    primary key(Aid)
);
```

### Functional Dependencies:

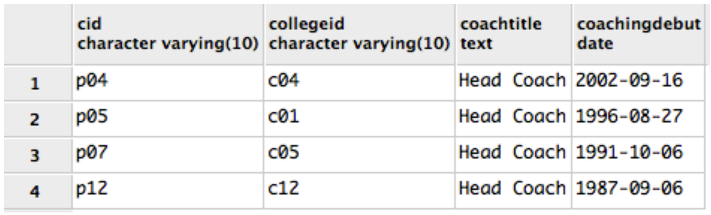
Aid → CollegeID, HeightInches, WeightPounds, GPA, CollegeStartDate



| | aid<br>character varying(10) | collegeid<br>character varying(10) | heightinches<br>bigint | weightpounds<br>bigint | gpa<br>numeric(3,2) | collegestartdate<br>date |
|---|---|---|---|---|---|---|
| 1 | p01 | c03 | 72 | 210 | 3.83 | 2014-09-01 |
| 2 | p02 | c02 | 71 | 205 | 3.12 | 2014-08-24 |
| 3 | p03 | c12 | 75 | 217 | 2.98 | 2013-09-04 |
| 4 | p05 | c03 | 67 | 170 | 4.00 | 2014-08-23 |
| 5 | p08 | c08 | 72 | 205 | 3.57 | 2014-08-15 |
| 6 | p09 | c05 | 70 | 220 | 3.18 | 2015-09-04 |
| 7 | p10 | c09 | 74 | 195 | 0.00 | 2016-08-25 |
| 8 | p11 | c06 | 66 | 165 | 3.79 | 2014-08-23 |
| 9 | p13 | c01 | 75 | 215 | 3.09 | 2013-08-15 |
| 10 | p14 | c07 | 70 | 220 | 3.00 | 2013-09-04 |
| 11 | p15 | c11 | 74 | 210 | 3.67 | 2014-08-25 |

## Coaches Table:

```
CREATE TABLE Coaches (
    Cid                 varchar(10) not null references Person(Pid),
    CollegeID           varchar(10) not null references CollegeTeams(CollegeID),
    CoachTitle          text,
    CoachingDebut       date,
  primary key(Cid)
);
```



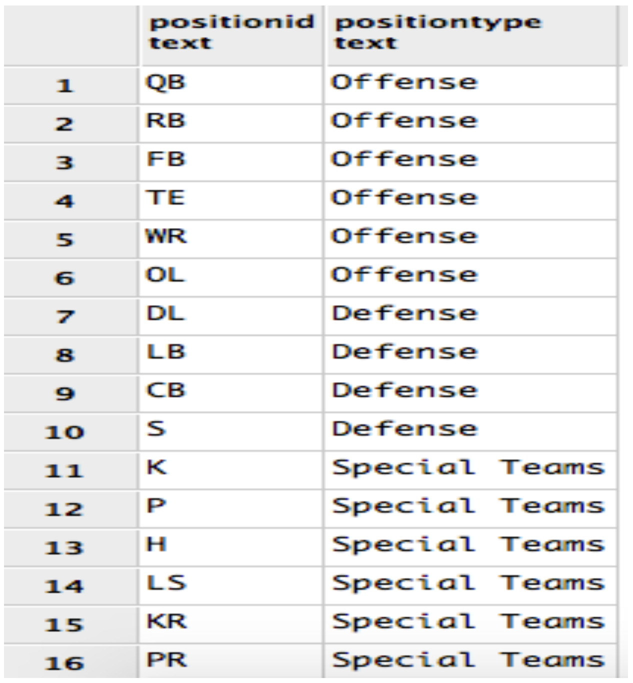
| | cid<br>character varying(10) | collegeid<br>character varying(10) | coachtitle<br>text | coachingdebut<br>date |
|---|---|---|---|---|
| 1 | p04 | c04 | Head Coach | 2002-09-16 |
| 2 | p05 | c01 | Head Coach | 1996-08-27 |
| 3 | p07 | c05 | Head Coach | 1991-10-06 |
| 4 | p12 | c12 | Head Coach | 1987-09-06 |

### Functional Dependencies:

Cid → CollegeID, CoachTitile, CoachingDebut

## Positions Table:

```
CREATE TABLE Positions (
      PositionID                    text unique not null,
      PositionType                  text,
  primary key(PositionID)
);
```



| | positionid text | positiontype text |
|---|---|---|
| 1 | QB | Offense |
| 2 | RB | Offense |
| 3 | FB | Offense |
| 4 | TE | Offense |
| 5 | WR | Offense |
| 6 | OL | Offense |
| 7 | DL | Defense |
| 8 | LB | Defense |
| 9 | CB | Defense |
| 10 | S | Defense |
| 11 | K | Special Teams |
| 12 | P | Special Teams |
| 13 | H | Special Teams |
| 14 | LS | Special Teams |
| 15 | KR | Special Teams |
| 16 | PR | Special Teams |

### Functional Dependencies:

PositionID → PositionType

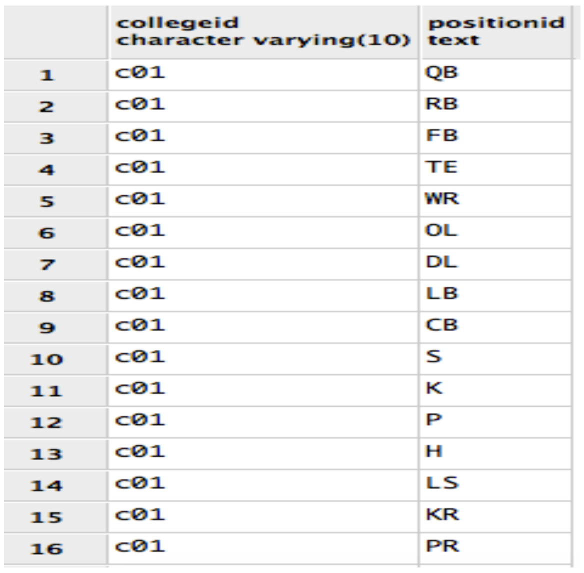
## TeamPositions Table:

This table is a weak entity that depends on the college team table and the position table. The purpose of this table is to show that many teams have many positions.

```
CREATE TABLE TeamPositions (
     CollegeID                 varchar(10) not null references CollegeTeams(CollegeID),
     PositionID                text not null references Positions(PositionID),
   primary key(CollegeID, PositionID)
);
```

### Functional Dependencies:

CollegeID, PositionID →



| | collegeid character varying(10) | positionid text |
|---|---|---|
| 1 | c01 | QB |
| 2 | c01 | RB |
| 3 | c01 | FB |
| 4 | c01 | TE |
| 5 | c01 | WR |
| 6 | c01 | OL |
| 7 | c01 | DL |
| 8 | c01 | LB |
| 9 | c01 | CB |
| 10 | c01 | S |
| 11 | c01 | K |
| 12 | c01 | P |
| 13 | c01 | H |
| 14 | c01 | LS |
| 15 | c01 | KR |
| 16 | c01 | PR |

**This is a snap shot of team c01. The other teams have the same positions as well.

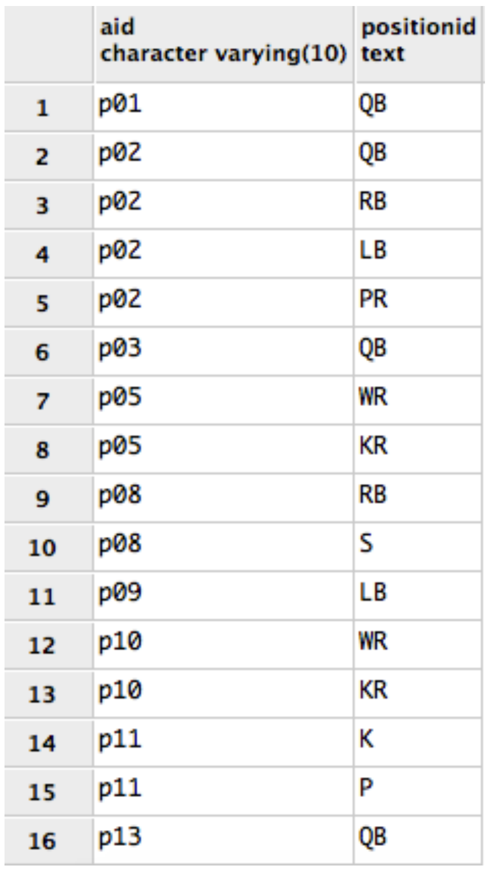
## AtheletePositions Table:

This table connects Athletes to the positions that they play. An athlete has the ability to play more than one position and multiple athletes can play the same position.

```
CREATE TABLE AthletePositions (
        Aid                      varchar(10) not null references Athletes(Aid),
        PositionID               text not null references Positions(PositionID),
  primary key (Aid, PositionID)
);
```

### Functional Dependencies:

Aid, PositionID →



| | aid<br>character varying(10) | positionid<br>text |
|---|---|---|
| 1 | p01 | QB |
| 2 | p02 | QB |
| 3 | p02 | RB |
| 4 | p02 | LB |
| 5 | p02 | PR |
| 6 | p03 | QB |
| 7 | p05 | WR |
| 8 | p05 | KR |
| 9 | p08 | RB |
| 10 | p08 | S |
| 11 | p09 | LB |
| 12 | p10 | WR |
| 13 | p10 | KR |
| 14 | p11 | K |
| 15 | p11 | P |
| 16 | p13 | QB |

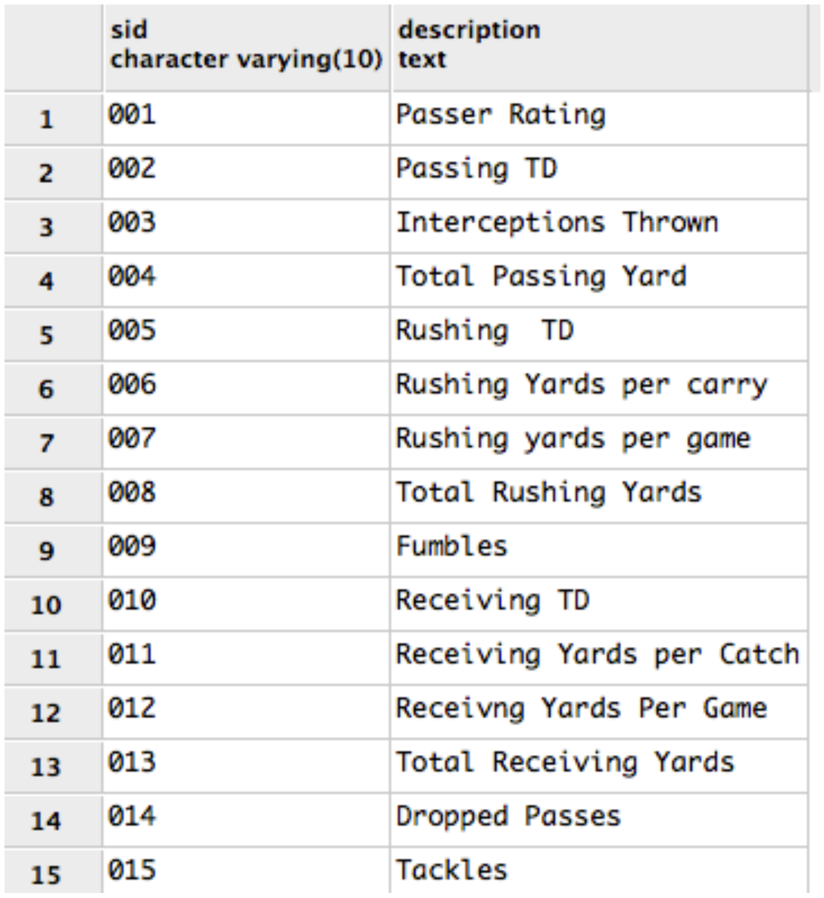
## StatsTable Table:

This is a general statistics table that contains a description related to each stat ID

```
CREATE TABLE StatsTable (
        Sid                 varchar(10) not null,
        Description         text,
  primary key(Sid)
);
```

### Functional Dependencies:

Sid → Description



|  | sid character varying(10) | description text |
|---|---|---|
| 1 | 001 | Passer Rating |
| 2 | 002 | Passing TD |
| 3 | 003 | Interceptions Thrown |
| 4 | 004 | Total Passing Yard |
| 5 | 005 | Rushing  TD |
| 6 | 006 | Rushing Yards per carry |
| 7 | 007 | Rushing yards per game |
| 8 | 008 | Total Rushing Yards |
| 9 | 009 | Fumbles |
| 10 | 010 | Receiving TD |
| 11 | 011 | Receiving Yards per Catch |
| 12 | 012 | Receivng Yards Per Game |
| 13 | 013 | Total Receiving Yards |
| 14 | 014 | Dropped Passes |
| 15 | 015 | Tackles |

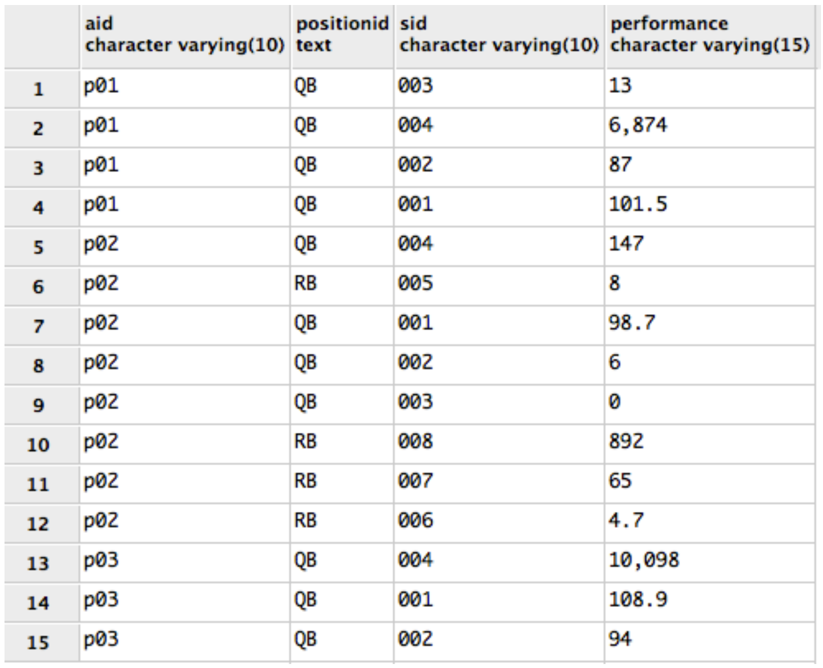
## **OffensiveStats Table:**

The purpose of this table is to only display stats that are relevant to players' offensive positions. This table dependant upon the athlete, the position he plays, and the stats table.

```
CREATE TABLE OffensiveStats (
        Aid                     varchar(10) not null references Athletes(Aid),
        PositionID              text not null references Positions(PositionID)
                                         check (PositionID = 'QB'   or PositionID = 'RB' or PositionID =
                                       'FB' or PositionID = 'WR' or PositionID = 'TE' or PositionID ='OL'),
       Sid                      varchar(10) not null references StatsTable(Sid),
       Performance              varchar(15),
  primary key(Sid, Aid, PositionID)
);
```

## Functional Dependencies:

<u>SID, AID, PositionID</u> → Performance



| | aid<br>character varying(10) | positionid<br>text | sid<br>character varying(10) | performance<br>character varying(15) |
|---|---|---|---|---|
| 1 | p01 | QB | 003 | 13 |
| 2 | p01 | QB | 004 | 6,874 |
| 3 | p01 | QB | 002 | 87 |
| 4 | p01 | QB | 001 | 101.5 |
| 5 | p02 | QB | 004 | 147 |
| 6 | p02 | RB | 005 | 8 |
| 7 | p02 | QB | 001 | 98.7 |
| 8 | p02 | QB | 002 | 6 |
| 9 | p02 | QB | 003 | 0 |
| 10 | p02 | RB | 008 | 892 |
| 11 | p02 | RB | 007 | 65 |
| 12 | p02 | RB | 006 | 4.7 |
| 13 | p03 | QB | 004 | 10,098 |
| 14 | p03 | QB | 001 | 108.9 |
| 15 | p03 | QB | 002 | 94 |

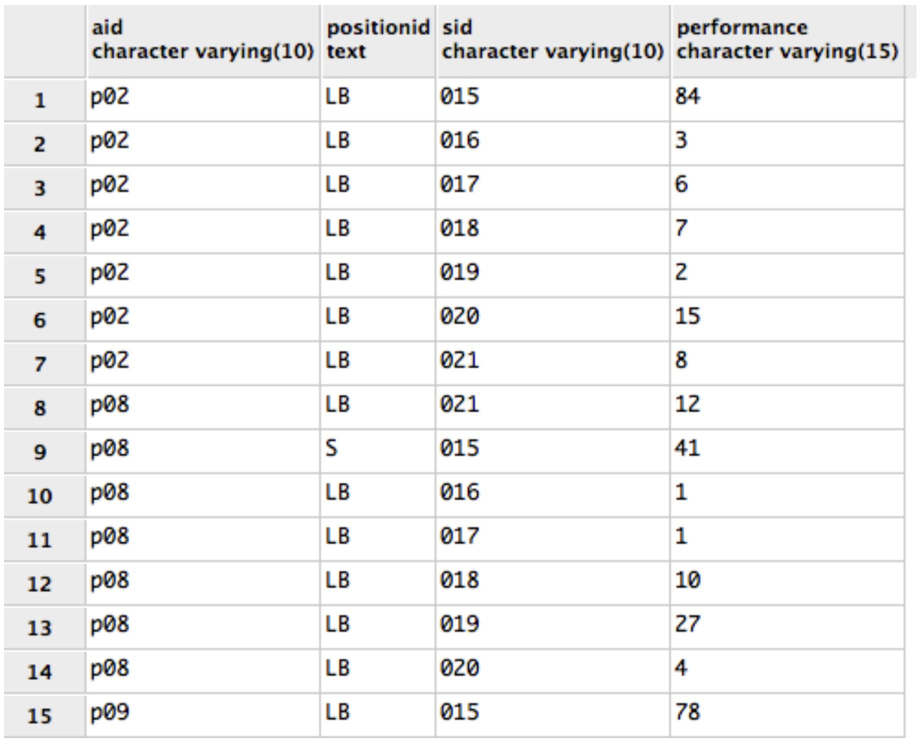
## DefensiveStats Table:

The purpose of this table is to only display stats that are relevant to players' defensive positions. This table dependent upon the athlete, the position he plays, and the stats table.

```
CREATE TABLE DefensiveStats (
    Aid                    varchar(10) not null references Athletes(Aid),
    PositionID             text not null references Positions(PositionID)
                                     check (PositionID = 'DL' or
                                     PositionID = 'LB' or PositionID = 'CB' or PositionID = 'S'),
    Sid                    varchar(10) not null references StatsTable(Sid),
    Performance            varchar(15),
  primary key(Sid, Aid, PositionID)
);
```

### Functional Dependencies:

<u>SID, AID, PositionID</u> → Performance



| | aid character varying(10) | positionid text | sid character varying(10) | performance character varying(15) |
|---|---|---|---|---|
| 1 | p02 | LB | 015 | 84 |
| 2 | p02 | LB | 016 | 3 |
| 3 | p02 | LB | 017 | 6 |
| 4 | p02 | LB | 018 | 7 |
| 5 | p02 | LB | 019 | 2 |
| 6 | p02 | LB | 020 | 15 |
| 7 | p02 | LB | 021 | 8 |
| 8 | p08 | LB | 021 | 12 |
| 9 | p08 | S | 015 | 41 |
| 10 | p08 | LB | 016 | 1 |
| 11 | p08 | LB | 017 | 1 |
| 12 | p08 | LB | 018 | 10 |
| 13 | p08 | LB | 019 | 27 |
| 14 | p08 | LB | 020 | 4 |
| 15 | p09 | LB | 015 | 78 |

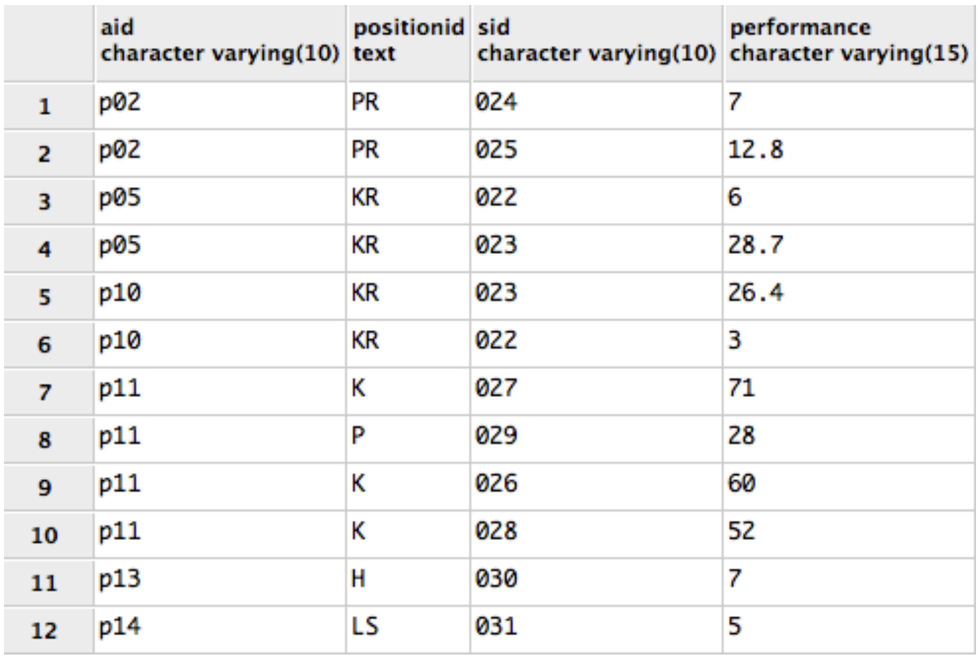
## SpecialTeamsStats Table:

The purpose of this table is to only display stats that are relevant to players' special teams positions.  This table dependent upon the athlete, the position he plays, and the stats table.

```
CREATE TABLE SpecialTeamsStats (
     Aid                    varchar(10) not null references Athletes(Aid),
     PositionID          text not null references Positions(PositionID)
                                check (PositionID = 'K' or     PositionID = 'P' or PositionID = 'H'
                                            or PositionID = 'LS' or PositionID = 'KR' or PositionID = 'PR'),
     Sid                    varchar(10) not null references StatsTable(Sid),
     Performance        varchar(15),
  primary key(Sid, Aid, PositionID)
);
```

### Functional Dependencies:

SID, AID, PositionID → Performance



| | aid character varying(10) | positionid text | sid character varying(10) | performance character varying(15) |
|---|---|---|---|---|
| 1 | p02 | PR | 024 | 7 |
| 2 | p02 | PR | 025 | 12.8 |
| 3 | p05 | KR | 022 | 6 |
| 4 | p05 | KR | 023 | 28.7 |
| 5 | p10 | KR | 023 | 26.4 |
| 6 | p10 | KR | 022 | 3 |
| 7 | p11 | K | 027 | 71 |
| 8 | p11 | P | 029 | 28 |
| 9 | p11 | K | 026 | 60 |
| 10 | p11 | K | 028 | 52 |
| 11 | p13 | H | 030 | 7 |
| 12 | p14 | LS | 031 | 5 |

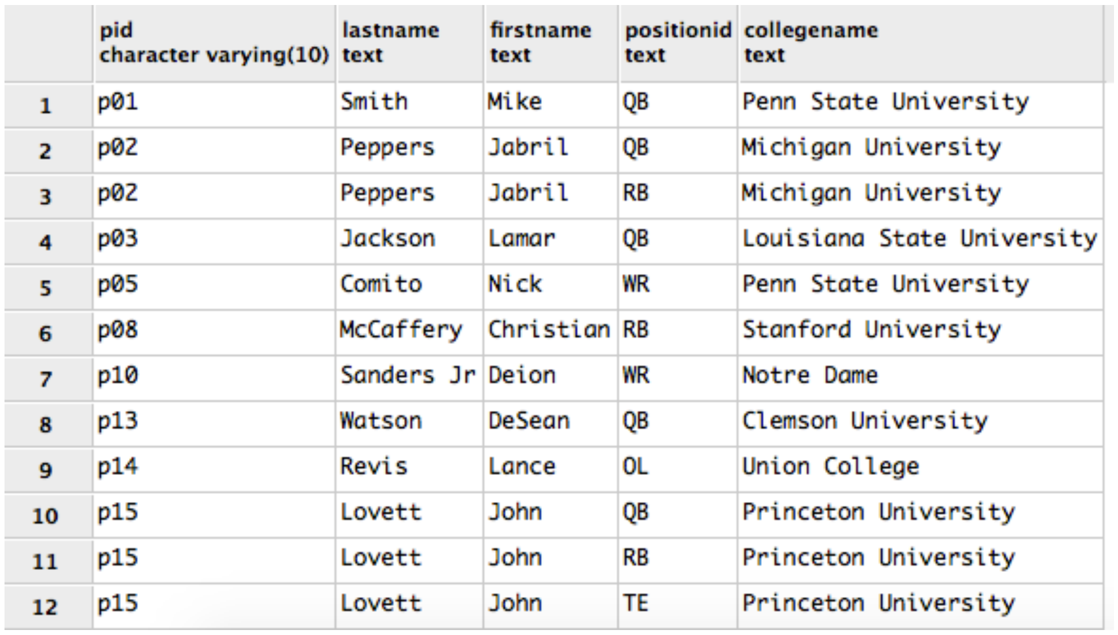
# Views

## Offensive Athletes Views:

The purpose of this view is for coaches and organizations looking to draft a player on the offensive side of the ball or just wants to track players who play offensive positions for future drafts.

```
CREATE VIEW OffensiveAthletes AS
select distinct p.pid, p.lastName, p.firstName, po.positionID, ct.collegeName
from athletes a, athletepositions ap, person p, positions po, teampositions tp, collegeTeams ct
where po.positionId = tp.positionID
        and tp.positionID = ap.positionID
        and ap.aid = a.aid
        and a.aid = p.pid
        and ct.collegeID = a.collegeID
        and po.positionType = 'Offense'
order by pid ASC;

select *
from offensiveAthletes
```



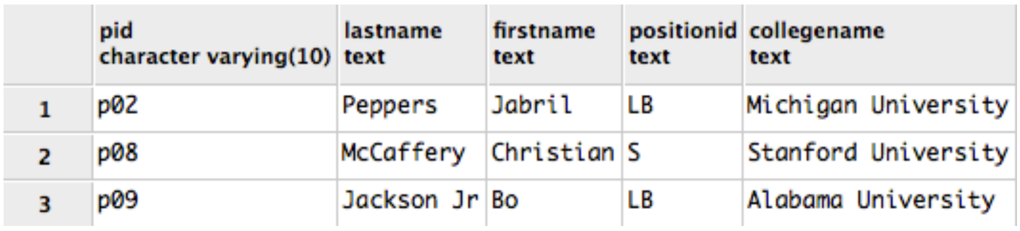
| | pid character varying(10) | lastname text | firstname text | positionid text | collegename text |
|---|---|---|---|---|---|
| 1 | p01 | Smith | Mike | QB | Penn State University |
| 2 | p02 | Peppers | Jabril | QB | Michigan University |
| 3 | p02 | Peppers | Jabril | RB | Michigan University |
| 4 | p03 | Jackson | Lamar | QB | Louisiana State University |
| 5 | p05 | Comito | Nick | WR | Penn State University |
| 6 | p08 | McCaffery | Christian | RB | Stanford University |
| 7 | p10 | Sanders Jr | Deion | WR | Notre Dame |
| 8 | p13 | Watson | DeSean | QB | Clemson University |
| 9 | p14 | Revis | Lance | OL | Union College |
| 10 | p15 | Lovett | John | QB | Princeton University |
| 11 | p15 | Lovett | John | RB | Princeton University |
| 12 | p15 | Lovett | John | TE | Princeton University |

## Defensive Athletes View:

The purpose of this view is to look at players who play on the defensive side of the ball.

```
CREATE VIEW DefensiveAthletes AS
select distinct p.pid, p.lastName, p.firstName, po.positionID, ct.collegeName
from athletes a, athletepositions ap, person p, positions po, teampositions tp, collegeTeams ct
where po.positionId = tp.positionID
        and tp.positionID = ap.positionID
        and ap.aid = a.aid
        and a.aid = p.pid
        and ct.collegeID = a.collegeID
        and po.positionType = 'Defense'
order by pid ASC;
```

```
select *
from defensiveAthletes
```



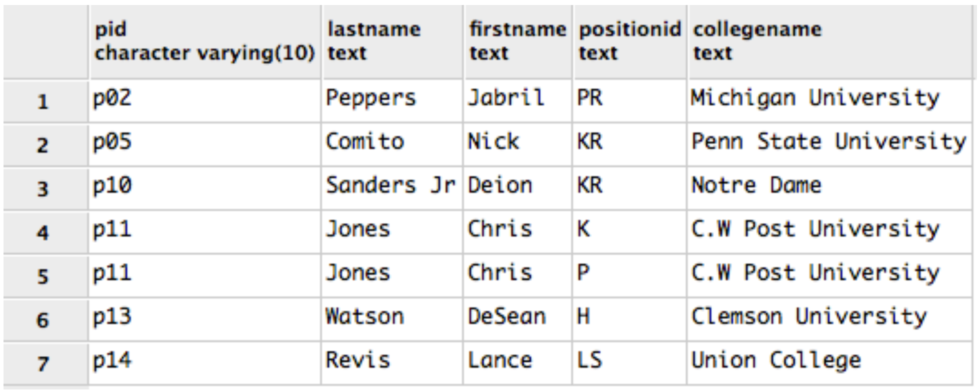
| | pid character varying(10) | lastname text | firstname text | positionid text | collegename text |
|---|---|---|---|---|---|
| 1 | p02 | Peppers | Jabril | LB | Michigan University |
| 2 | p08 | McCaffery | Christian | S | Stanford University |
| 3 | p09 | Jackson Jr | Bo | LB | Alabama University |

## Special Teams Athletes

The purpose of this view is to show the athletes who play on Special Teams.

```
CREATE VIEW SpecialTeamsAthletes As
select distinct p.pid, p.lastName, p.firstName, po.positionID, ct.collegeName
from athletes a, athletepositions ap, person p, positions po, teampositions tp, collegeTeams ct
where po.positionId = tp.positionID
        and tp.positionID = ap.positionID
        and ap.aid = a.aid
        and a.aid = p.pid
        and ct.collegeID = a.collegeID
        and po.positionType = 'Special Teams'
order by pid ASC;
```

```
select *
from SpecialTeamsAthletes
```



| | pid character varying(10) | lastname text | firstname text | positionid text | collegename text |
|---|---|---|---|---|---|
| 1 | p02 | Peppers | Jabril | PR | Michigan University |
| 2 | p05 | Comito | Nick | KR | Penn State University |
| 3 | p10 | Sanders Jr | Deion | KR | Notre Dame |
| 4 | p11 | Jones | Chris | K | C.W Post University |
| 5 | p11 | Jones | Chris | P | C.W Post University |
| 6 | p13 | Watson | DeSean | H | Clemson University |
| 7 | p14 | Revis | Lance | LS | Union College |

# Stored Procedures

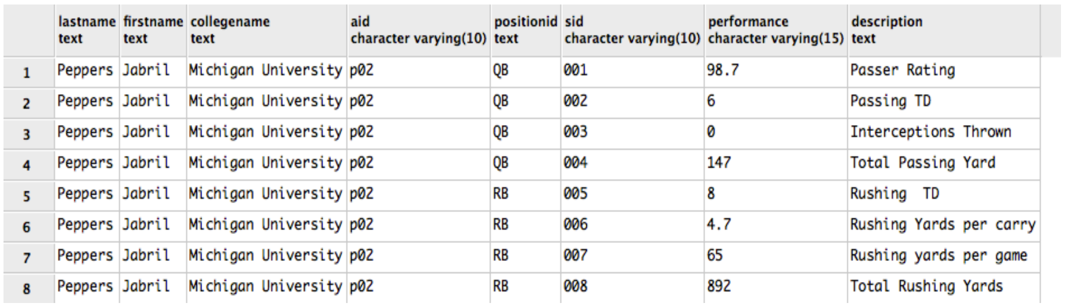
## OffenseStatsFor:

The purpose of this stored procedure is to make in convenient for a scout, coach, or organization to look up the offensive stats for a particular player by entering their last name followed by their first name rather than having to filter through each offensive player. The LIKE command was used instead of equal for p.lastName and p.firstName when comparing it to AthleteLastName and AthleteFirstName for user purposes incase the user does not know exactly how to spell a players name.

**Defensive Stats and Special Teams stats can be found using a similar store procedure

```
create or replace function OffenseStatsFor(text, text,REFCURSOR) returns refcursor as
$$
declare
        AthleteLastName   text          := $1;
        AthleteFirstName  text          := $2;
        resultset       REFCURSOR       := $3;
begin
    open resultset for
        select distinct p.lastname,p.firstname, ct.collegeName, o.*, st.description
        from statsTable st, offensiveStats o, person p, collegeTeams ct, athletes a
        where o.aid = p.pid
            and p.lastName LIKE AthleteLastName
            and p.firstName LIKE AthleteFirstName
            and o.sid = st.sid
            and o.aid = a.aid
            and a.aid = p.pid
            and a.collegeID = ct.collegeID
            order by positionID ASC;
     return resultset;
end;
$$
language plpgsql;

select OffenseStatsFor('P%', 'J%','results');
fetch all from results;
```



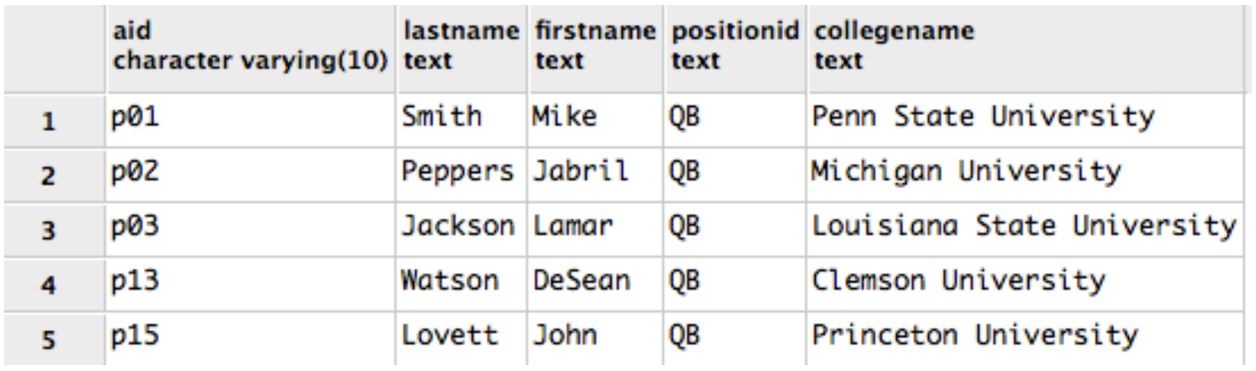
| | lastname text | firstname text | collegename text | aid character varying(10) | positionid text | sid character varying(10) | performance character varying(15) | description text |
|---|---|---|---|---|---|---|---|---|
| 1 | Peppers | Jabril | Michigan University | p02 | QB | 001 | 98.7 | Passer Rating |
| 2 | Peppers | Jabril | Michigan University | p02 | QB | 002 | 6 | Passing TD |
| 3 | Peppers | Jabril | Michigan University | p02 | QB | 003 | 0 | Interceptions Thrown |
| 4 | Peppers | Jabril | Michigan University | p02 | QB | 004 | 147 | Total Passing Yard |
| 5 | Peppers | Jabril | Michigan University | p02 | RB | 005 | 8 | Rushing  TD |
| 6 | Peppers | Jabril | Michigan University | p02 | RB | 006 | 4.7 | Rushing Yards per carry |
| 7 | Peppers | Jabril | Michigan University | p02 | RB | 007 | 65 | Rushing yards per game |
| 8 | Peppers | Jabril | Michigan University | p02 | RB | 008 | 892 | Total Rushing Yards |

## Get_Athlete_by_Position

This stored procedure is again meant for the convenience of scouts, coaches and organizations. For example, in the upcoming draft, the New York Jets may be looking to draft a quarterback or track a quarterback for future drafts to come. This store procedure allows users to look up players by inputting a position.

```
create or replace function Get_Athlete_by_Position(text,REFCURSOR) returns refcursor as
$$
declare
    Position    text      :=  $1;
    resultset   REFCURSOR  := $2;
begin
   open resultset for
        select distinct ap.aid, p.firstName, p.lastName, ap.positionID, ct.collegeName
        from AthletePositions ap, person p, collegeTeams ct, athletes a
        where a.collegeID = ct.collegeID
            and ap.aid = a.aid
            and a.aid = p.pid
            and ap.positionID = Position
       order by aid ASC;
   return resultset;
end;
$$
language plpgsql;

select Get_Athlete_by_Position('QB', 'results2');
fetch all from results2;
```



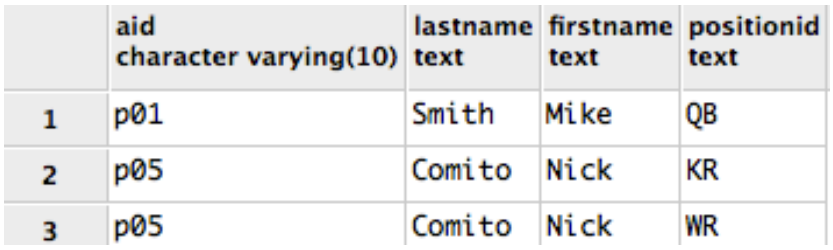
| | aid<br>character varying(10) | lastname<br>text | firstname<br>text | positionid<br>text | collegename<br>text |
|---|---|---|---|---|---|
| 1 | p01 | Smith | Mike | QB | Penn State University |
| 2 | p02 | Peppers | Jabril | QB | Michigan University |
| 3 | p03 | Jackson | Lamar | QB | Louisiana State University |
| 4 | p13 | Watson | DeSean | QB | Clemson University |
| 5 | p15 | Lovett | John | QB | Princeton University |

## Get Athlete by College

The purpose of this stored procedure is to make it easy for users to look up players that play for a particular college. This will be useful for NFL scouts, coaches, and organizations who like either the style of offense or defense a particular college runs and believes a player from that college could fit into the system their NFL team uses. This is particularly useful for teams looking at quarterbacks since the style of offense between college and the NFL is very different, while some colleges use a more pro style then others.

```
create or replace function Get_Athlete_by_College(text,REFCURSOR) returns refcursor as
$$
declare
    CollegeNamePassIn   text            :=  $1;
    resultset           REFCURSOR  :=  $2;
begin
    open resultset for
        select distinct ap.aid, p.lastName, p.firstName, ap.positionID
        from AthletePositions ap, person p, collegeTeams ct, athletes a
        where a.collegeID = ct.collegeID
            and ap.aid = a.aid
            and a.aid = p.pid
            and ct.collegeName = CollegeNamePassIn
        order by aid ASC;
   return resultset;
end;
$$
language plpgsql;

select Get_Athlete_by_College('Penn State University', 'results3');
fetch all from results3;
```



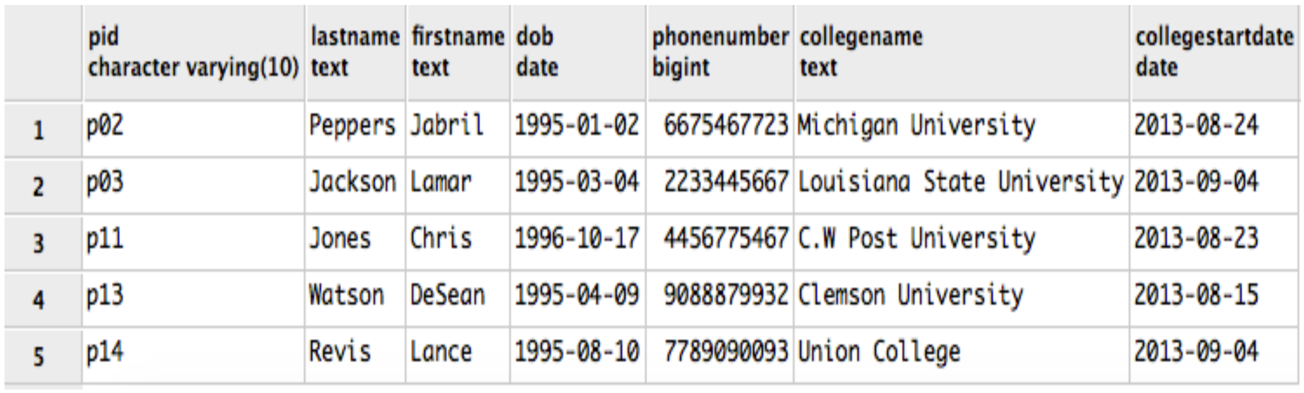
| | aid<br>character varying(10) | lastname<br>text | firstname<br>text | positionid<br>text |
|---|---|---|---|---|
| 1 | p01 | Smith | Mike | QB |
| 2 | p05 | Comito | Nick | KR |
| 3 | p05 | Comito | Nick | WR |

## Reports

### Draft Eligible Athletes:

The purpose of generating this report is to see which players in the database are eligible for the upcoming NFL Draft. College football athletes, under NCAA rules, must be enrolled in college for 3 years before being draft eligible.

```
select p.*, ct.collegeName, a.collegeStartDate
from person p, athletes a, collegeTeams ct
where pid = aid
    and ct.collegeID = a.collegeID
    and extract (day from now() - a.collegeStartDate) > 365 * 3;
```

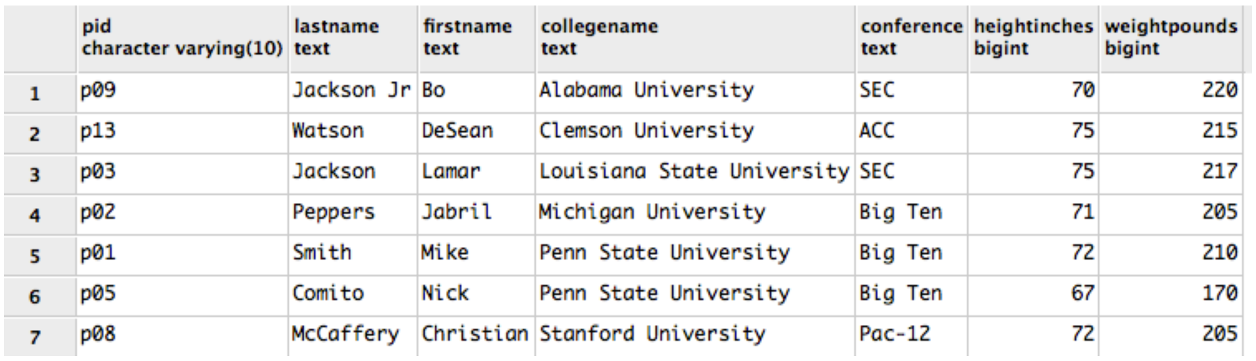


| | pid<br>character varying(10) | lastname<br>text | firstname<br>text | dob<br>date | phonenumber<br>bigint | collegename<br>text | collegestartdate<br>date |
|---|---|---|---|---|---|---|---|
| 1 | p02 | Peppers | Jabril | 1995-01-02 | 6675467723 | Michigan University | 2013-08-24 |
| 2 | p03 | Jackson | Lamar | 1995-03-04 | 2233445667 | Louisiana State University | 2013-09-04 |
| 3 | p11 | Jones | Chris | 1996-10-17 | 4456775467 | C.W Post University | 2013-08-23 |
| 4 | p13 | Watson | DeSean | 1995-04-09 | 9088879932 | Clemson University | 2013-08-15 |
| 5 | p14 | Revis | Lance | 1995-08-10 | 7789090093 | Union College | 2013-09-04 |

## Athletes in the Big Five

The purpose of this report is to show athletes that play in one of the Big Five conferences. These conferences are part of the Division I level and are considered the most powerful and competitive conferences in the country. They are the SEC, ACC, Big Ten, Big 12, and Pac-12. This report may be useful for NFL teams looking for top tier players based on the level of competition the player faces during his college career.

```
select distinct p.pid, p.lastName, p.firstName, ct.collegeName, ct.conference,
                  a.heightInches, a.weightPounds
from athletes a, collegeTeams ct, person p
where p.pid = a.aid
     and ct.collegeID = a.collegeID
     and ct.collegeName IN (select distinct ct.collegeName
                        from collegeTeams ct
                        where ct.NCAADivision = 1
                         and ct.conference IN ('SEC', 'ACC', 'Big Ten', 'Pac-12', 'Big 12')
                    )
Order by ct.collegeName
```



| | pid character varying(10) | lastname text | firstname text | collegename text | conference text | heightinches bigint | weightpounds bigint |
|---|---|---|---|---|---|---|---|
| 1 | p09 | Jackson Jr | Bo | Alabama University | SEC | 70 | 220 |
| 2 | p13 | Watson | DeSean | Clemson University | ACC | 75 | 215 |
| 3 | p03 | Jackson | Lamar | Louisiana State University | SEC | 75 | 217 |
| 4 | p02 | Peppers | Jabril | Michigan University | Big Ten | 71 | 205 |
| 5 | p01 | Smith | Mike | Penn State University | Big Ten | 72 | 210 |
| 6 | p05 | Comito | Nick | Penn State University | Big Ten | 67 | 170 |
| 7 | p08 | McCaffery | Christian | Stanford University | Pac-12 | 72 | 205 |

## Security

There are three user groups regarding this database. They include the NCAA database administrators that run and maintain the database, college teams that use the database for studying opponents and recruiting potential transfers, and NFL teams that use the database to make decisions regarding the NFL Draft and free agent signings. The NCAA administrators are the only users that are allowed to insert into and update the database. If college teams need to make a change regarding their players, coaches, or stats, they need to send the request to the NCAA in order to avoid NCAA violations and tampering with official statistics. The NCAA also updates stats after every game when the colleges submit them. With that being said, college teams are only able to view (or select) tables in the database. NFL teams are only allowed to view (or select) tables since they are only using the database as a reference to make decisions.

**NCAA Admin Role:**

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

**College Admin Role**

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

**NFL Teams Role**

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

## Implementation Notes

This database was created and implemented using PostgresSQL version 9.3. Overall, implementing this database was fairly smooth. The main trouble came with deciding on how to organize stats in a clear and logical manner. For the purpose of this database I chose to make just one table relating to each type of position (offense, defense, special teams) rather than making a stats table for each individual position. A general stats table was necessary in order to give descriptions for each stat. Another issue I faced was making sure a player who plays an offensive position ends up having stats in the appropriate table. In order to do this I was forced to put check constraints on OffensiveStats, DefensiveStats, and SpecialTeamsStats tables. The position ID needed to correspond the correct type of position in order for the stats to be inserted into the table.

## Known Issues

Like mentioned above, stat implementation caused problems when creating the database. The current way that the stats are organized does not allow a user to see a particular stat for a person without viewing every stat that pertains to that athlete in his respected category (offense, defense, special teams). For example, if a user wanted to look up a particular quarterback stat for an athlete, every quarterback stat related to that player will appear. This is because there are not individual stat tables pertaining to each individual position. Another issue is the database does not show if a player is a current athlete or an athlete that is no longer a member of the NCAA. A user cannot delete an athlete if they are no longer active since they would be destroying historical data and statistics.

## Future Enhancements

As I continue to improve upon this database, I will need to observe how the database is functioning before making any drastic changes. Some future enhancements I have in mind already are to improve upon the statistics and separate offensive, defensive, and special teams stats even further. The database currently does not account for players who transfer. I will need to develop a way to track the stats an athlete had at one school as well as the stats he will obtain at another while still maintaining referential integrity throughout the database. Currently the database displays career statistics that accumulate as the NCAA administrators update them. In the future I would like to expand upon the database and be able to divide a players stats on a season-by-season basis as well as a cumulative career basis.