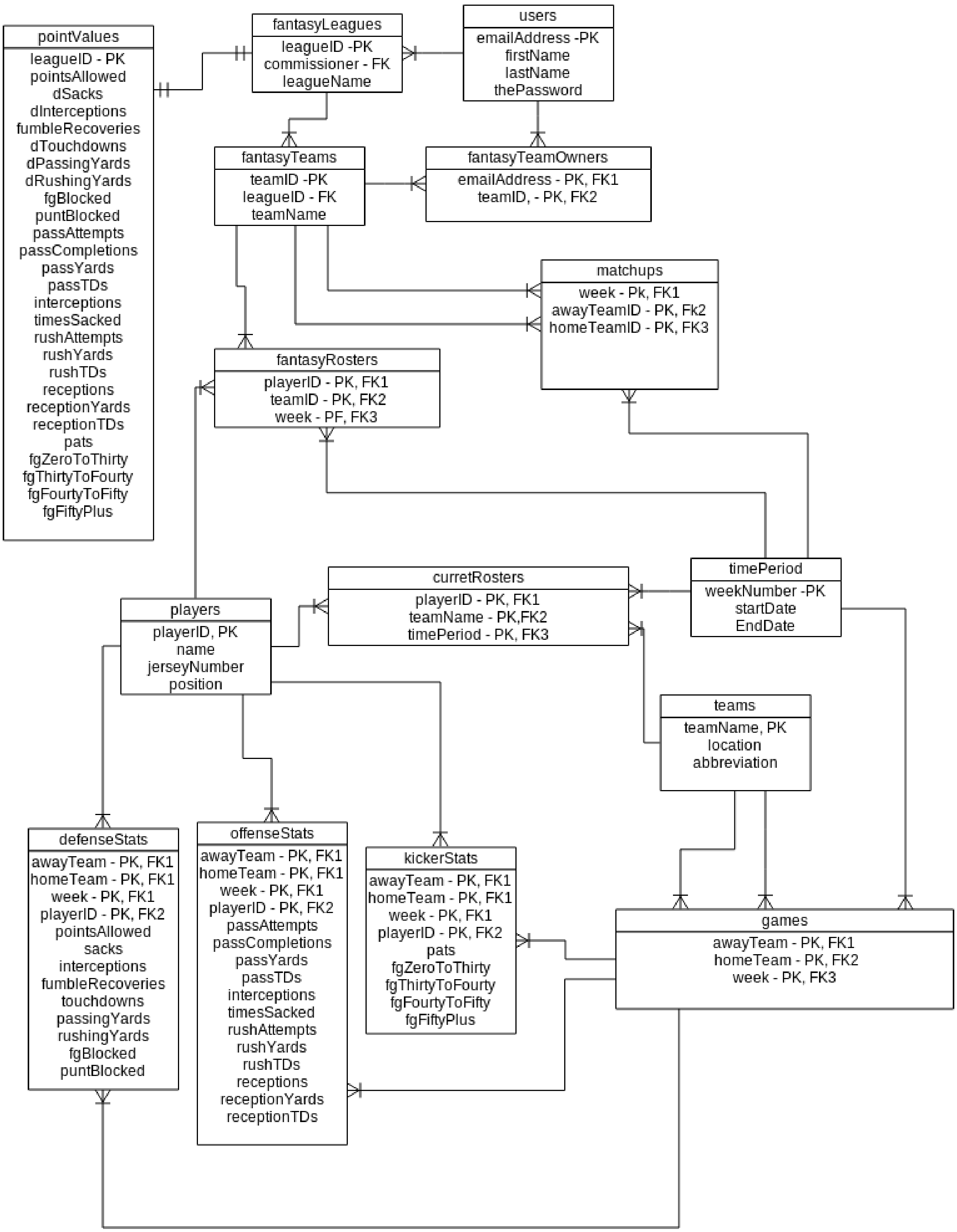
Fantasy Football Database

Designed by Ryan Bertsche

April 25, 2014

Table of Contents

Entity Relationship Diagram



Executive Summary

Overview

Fantasy football is quickly becoming an extremely popular game that is commonly played on the internet. The game involves creating a team of football players, and play other teams head to head. Teams are organized into leagues, and teams in that league compete against each other. This requires a database that stores the roster of each team, the stats of the players on that team, as well the match ups between teams in a league.

Objectives

The purpose of this database is to store all information that is required to operate a complete fantasy football service. This database will be able to store the information about the NFL teams, all the players and the games they play. Each player will be associated with statistics for each game. That establishes the information needed to give teams scores. There are then tables for the different leagues, teams, and rosters of each team. All together, this database provides everything needed for a fantasy football service.

Tables

Users

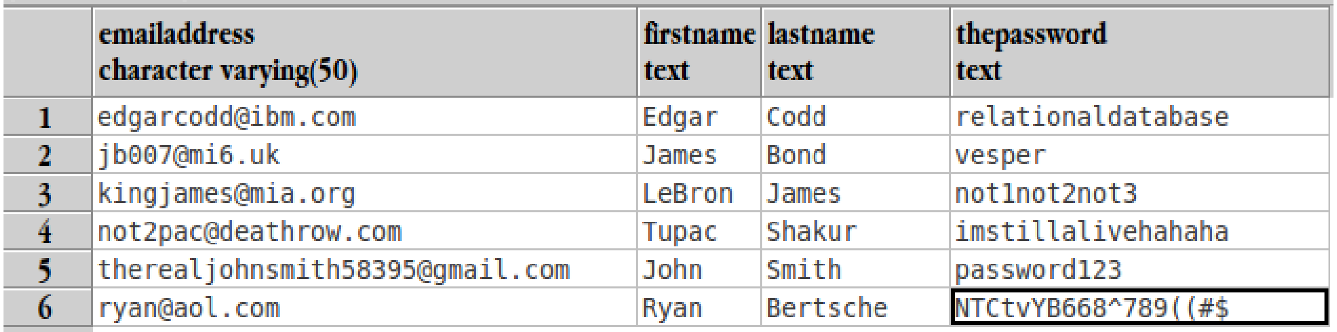
Purpose: Stores personal information for users based on their email addresses.

Functional Dependencies: emailAddress → firstName, lastName, thePassword

Create Statement:

```
CREATE TABLE users
(
        emailAddress  varchar(50)   NOT NULL PRIMARY KEY,
        firstName            text                  NOT NULL,
        lastName             text                  NOT NULL,
        thePassword          text                  NOT NULL
);
```

Sample Data:



	emailaddress character varying(50)	firstname text	lastname text	thepassword text
1	edgarcodd@ibm.com	Edgar	Codd	relationaldatabase
2	jb007@mi6.uk	James	Bond	vesper
3	kingjames@mia.org	LeBron	James	not1not2not3
4	not2pac@deathrow.com	Tupac	Shakur	imstillalivehahaha
5	therealjohnsmith58395@gmail.com	John	Smith	password123
6	ryan@aol.com	Ryan	Bertsche	NTCtvYB668^789((#$

FantasyLeagues

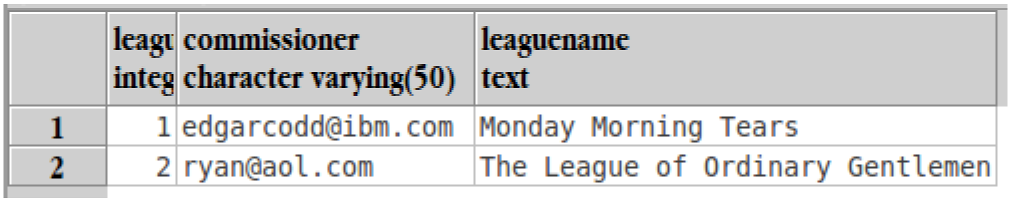
Purpose: Holds the ID and name of every league, along with the commissioner of the league

Functional Dependencies: leagueID → commissioner, leagueName

Create Statement:

```
CREATE TABLE fantasyLeagues
(
        leagueID          int                          NOT NULL PRIMARY KEY,
        commissioner varchar(50)          NOT NULL REFERENCES users(emailAddress),
        leagueName        text                     NOT NULL
);
```

Sample Data:



	leagu integ	commissioner character varying(50)	leaguename text
1	1	edgarcodd@ibm.com	Monday Morning Tears
2	2	ryan@aol.com	The League of Ordinary Gentlemen

FantasyTeams

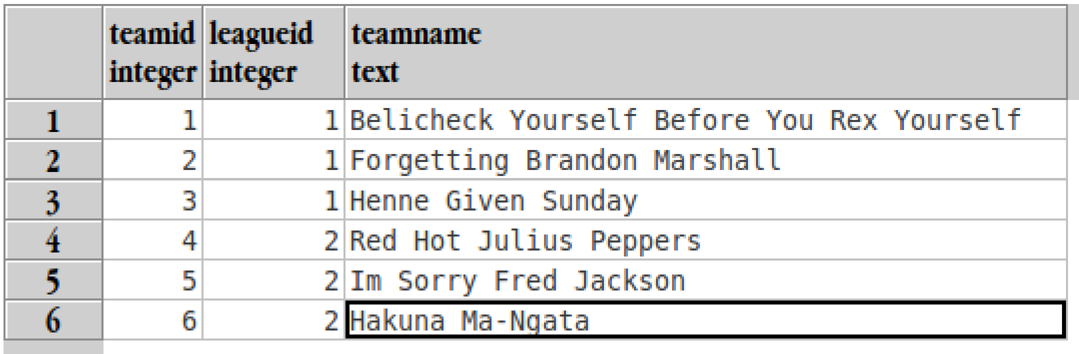
Purpose: Establishes teamID, associates with a league, and holds the name of the fantasy team

Functional Dependencies: (teamID, leagueID) → teamName

Create Statement:

```
CREATE TABLE fantasyTeams
(
        teamID              int             NOT NULL PRIMARY KEY,
        leagueID            int             NOT NULL REFERENCES fantasyLeagues(leagueID),
        teamName            text            NOT NULL
);
```

Sample Data:



	teamid integer	leagueid integer	teamname text
1	1	1	Belicheck Yourself Before You Rex Yourself
2	2	1	Forgetting Brandon Marshall
3	3	1	Henne Given Sunday
4	4	2	Red Hot Julius Peppers
5	5	2	Im Sorry Fred Jackson
6	6	2	Hakuna Ma-Ngata

FantasyTeamOwners

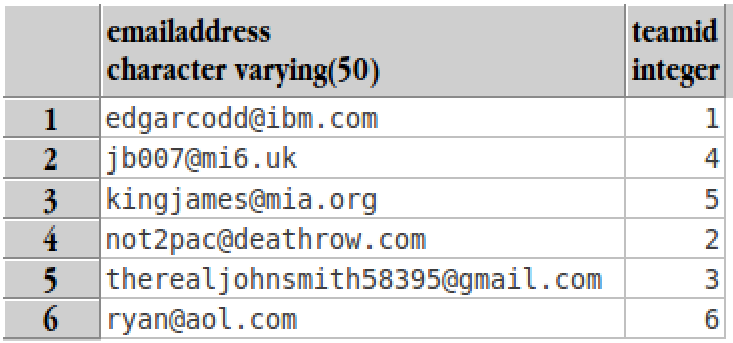
Purpose: Associates users with their teams

Functional Dependencies: (emailAddress, teamID) →

Create Statement:

```
CREATE TABLE fantasyTeamOwners
(
        emailAddress        varchar(50)     NOT NULL REFERENCES users(emailAddress),
        teamID              int             NOT NULL REFERENCES fantasyTeams(teamID),
PRIMARY KEY(emailAddress, teamID)
);
```

Sample Data:



	emailaddress character varying(50)	teamid integer
1	edgarcodd@ibm.com	1
2	jb007@mi6.uk	4
3	kingjames@mia.org	5
4	not2pac@deathrow.com	2
5	therealjohnsmith58395@gmail.com	3
6	ryan@aol.com	6

Time Period

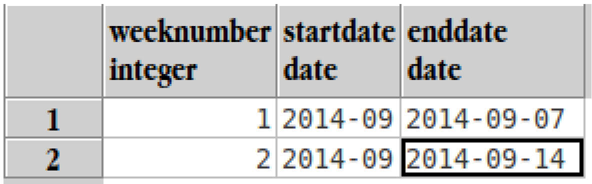
Purpose: Create time periods based on the weeks of the football season, to differentiate rosters and matchups, because they change every week.

Functional Dependencies: weekNumber → startDate, endDate

Create Statement:

```
CREATE TABLE timePeriod
(
        weekNumber          int                     NOT NULL PRIMARY KEY,
        startDate           date                    NOT NULL UNIQUE,
        endDate             date                    NOT NULL UNIQUE
);
```

Sample Data:



	weeknumber integer	startdate date	enddate date
1	1	2014-09	2014-09-07
2	2	2014-09	2014-09-14

Teams

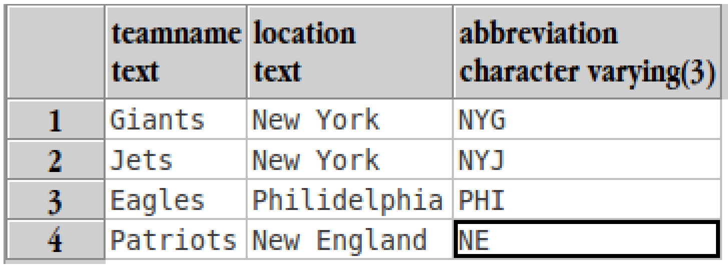
Purpose: Contains the NFL teams, names, as well as their location and abbreviated title

Functional Dependencies: teamName → location, abbreviation

Create Statement:

```
CREATE TABLE teams
(
        teamName            text                NOT NULL PRIMARY KEY,
        location            text                NOT NULL,
        abbreviation        varchar(3)          NOT NULL UNIQUE
);
```

Sample Data:



	teamname text	location text	abbreviation character varying(3)
1	Giants	New York	NYG
2	Jets	New York	NYJ
3	Eagles	Philidelphia	PHI
4	Patriots	New England	NE

Players

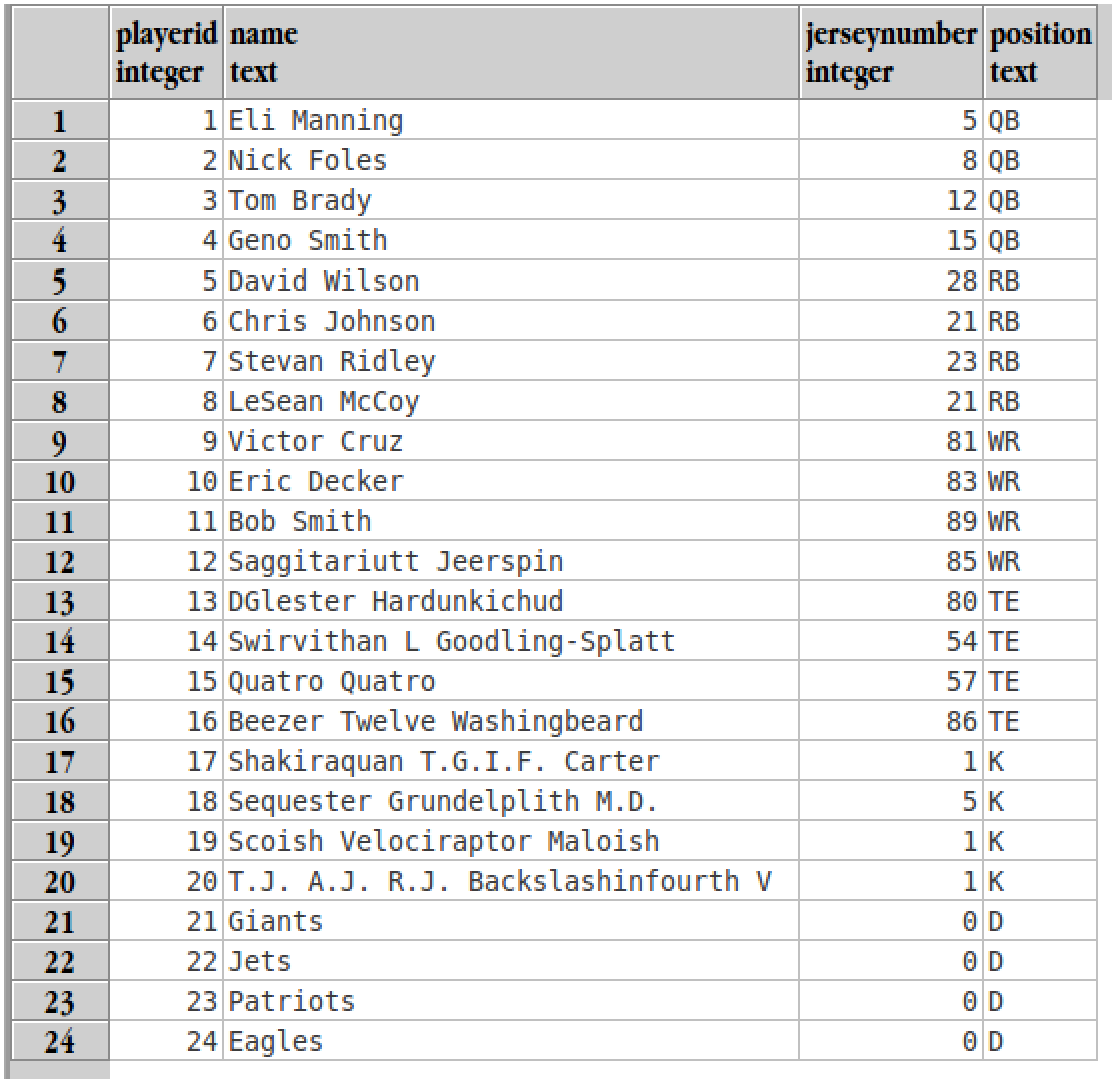
Purpose: This is meant to contain all of the players in the NFL, along with their name, position and jersey number.

Functional Dependencies: playerID → name, jerseyNumber, position

Create Statement:

```
CREATE TABLE players
(
        playerID             int                  NOT NULL PRIMARY KEY,
        name                 text                 NOT NULL,
        jerseyNumber         int                  NOT NULL,
        position             text                 NOT NULL CHECK
                (position in ('QB', 'WR', 'TE', 'RB', 'K', 'D'))
);
```

Sample Data:



	playerid integer	name text	jerseynumber integer	position text
1	1	Eli Manning	5	QB
2	2	Nick Foles	8	QB
3	3	Tom Brady	12	QB
4	4	Geno Smith	15	QB
5	5	David Wilson	28	RB
6	6	Chris Johnson	21	RB
7	7	Stevan Ridley	23	RB
8	8	LeSean McCoy	21	RB
9	9	Victor Cruz	81	WR
10	10	Eric Decker	83	WR
11	11	Bob Smith	89	WR
12	12	Saggitariutt Jeerspin	85	WR
13	13	DGlester Hardunkichud	80	TE
14	14	Swirvithan L Goodling-Splatt	54	TE
15	15	Quatro Quatro	57	TE
16	16	Beezer Twelve Washingbeard	86	TE
17	17	Shakiraquan T.G.I.F. Carter	1	K
18	18	Sequester Grundelplith M.D.	5	K
19	19	Scoish Velociraptor Maloish	1	K
20	20	T.J. A.J. R.J. Backslashinfourth V	1	K
21	21	Giants	0	D
22	22	Jets	0	D
23	23	Patriots	0	D
24	24	Eagles	0	D

Games

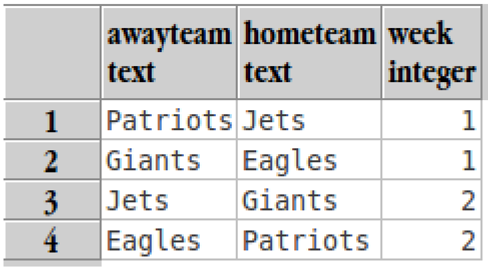
Purpose: Contains all the NFL games being played, based on the two teams playing and the week

Functional Dependencies: (awayTeam, homeTeam, week) →

Create Statement:

```
(
	awayTeam		text		NOT NULL REFERENCES teams(teamName),
	homeTeam		text		NOT NULL REFERENCES teams(teamName),
	week			int		NOT NULL REFERENCES timePeriod(weekNumber),
PRIMARY KEY(awayTeam, homeTeam, week)
);
```

Sample Data:



	awayteam text	hometeam text	week integer
1	Patriots	Jets	1
2	Giants	Eagles	1
3	Jets	Giants	2
4	Eagles	Patriots	2

DefenseStats

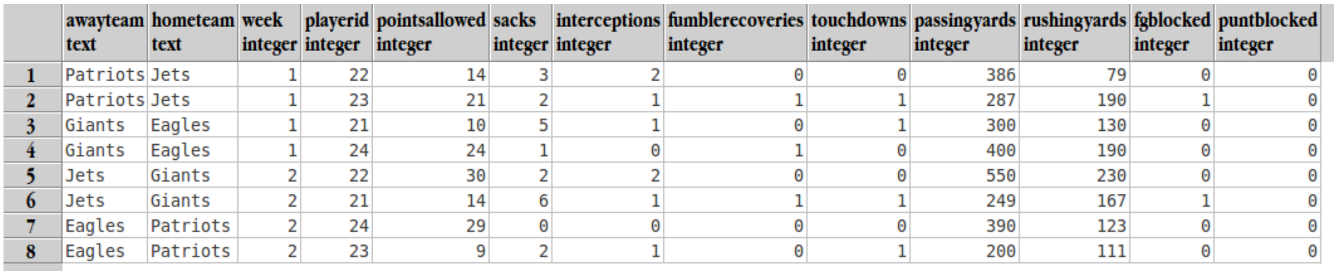
Purpose: Conatins the value of all defensive stats based on the game that was played and the player

Functional Dependencies: (awayTeam, homeTeam, week, playerID) → pointsAllowed, sack, interceptions, fumbleRecoveries, touchdowns, passingYards, RushingYards, fgBlocked, puntBlocked

Create Statement:

```
CREATE TABLE defenseStats
(
        awayTeam            text                    NOT NULL,
        homeTeam            text                    NOT NULL,
        week                int                             NOT NULL,
        playerID            int                             NOT NULL REFERENCES
players(playerID),
        pointsAllowedint                            NOT NULL DEFAULT 0,
        sacks               int                             NOT NULL DEFAULT 0,
        interceptions   int                         NOT NULL DEFAULT 0,
        fumbleRecoveries int                        NOT NULL DEFAULT 0,
        touchdowns          int                             NOT NULL DEFAULT 0,
        passingYards    int                         NOT NULL DEFAULT 0,
        rushingYards    int                         NOT NULL DEFAULT 0,
        fgBlocked           int                             NOT NULL DEFAULT 0,
        puntBlocked         int                             NOT NULL DEFAULT 0,
FOREIGN KEY(awayTeam, homeTeam, week) REFERENCES games(awayTeam, homeTeam, week),
PRIMARY KEY(awayTeam, homeTeam, week, playerID)
);
```

Sample Data:



	awayteam text	hometeam text	week integer	playerid integer	pointsallowed integer	sacks integer	interceptions integer	fumblerecoveries integer	touchdowns integer	passingyards integer	rushingyards integer	fgblocked integer	puntblocked integer
1	Patriots	Jets	1	22	14	3	2	0	0	386	79	0	0
2	Patriots	Jets	1	23	21	2	1	1	1	287	190	1	0
3	Giants	Eagles	1	21	10	5	1	0	1	300	130	0	0
4	Giants	Eagles	1	24	24	1	0	1	0	400	190	0	0
5	Jets	Giants	2	22	30	2	2	0	0	550	230	0	0
6	Jets	Giants	2	21	14	6	1	1	1	249	167	1	0
7	Eagles	Patriots	2	24	29	0	0	0	0	390	123	0	0
8	Eagles	Patriots	2	23	9	2	1	0	1	200	111	0	0

OffenseStats

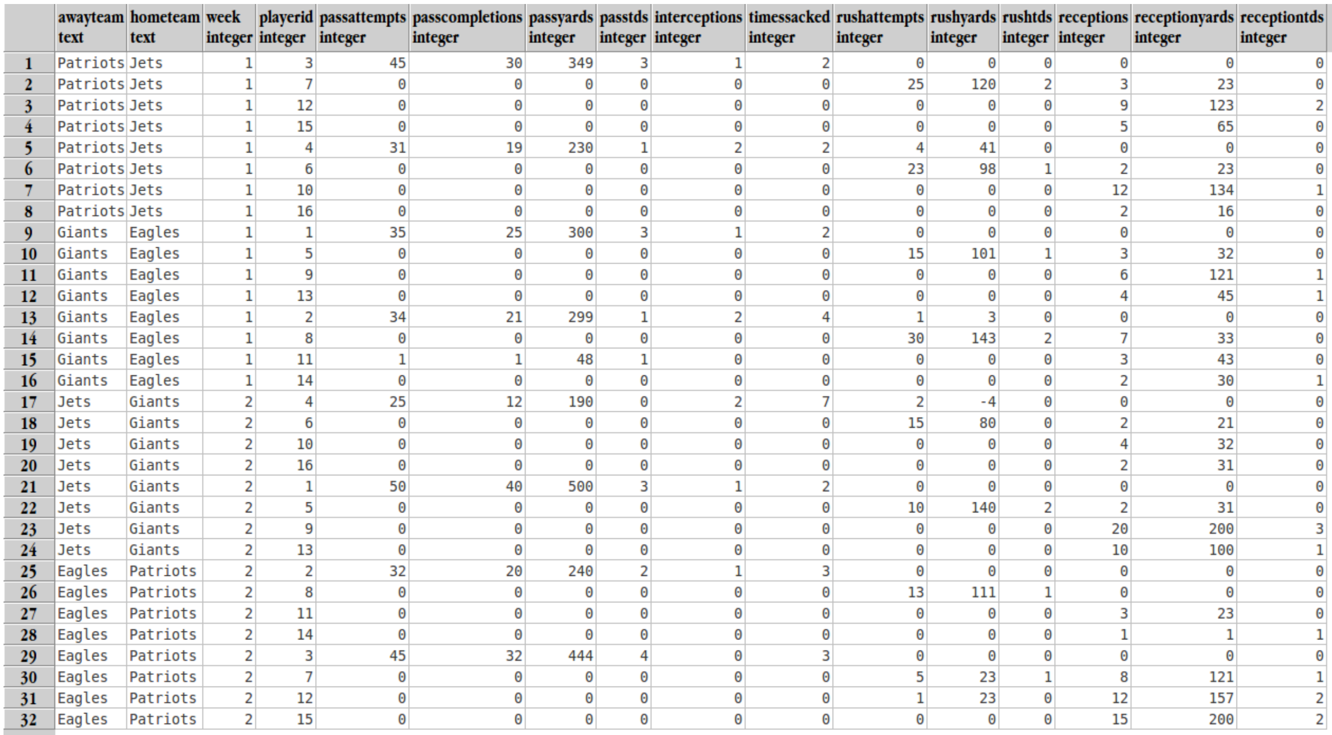
Purpose: Conatins the value of all offensive stats based on the game that was played and the player

Functional Dependencies: (awayTeam, homeTeam, week, playerID) → passAttempts, passCompletions, passYards, passTDs, interceptions, timesSacked, rushAttempts, rushYards, rushTDs, receptions, receptionYards, receptionTDs

Create Statement:

```
CREATE TABLE offenseStats
(
    awayTeam        text            NOT NULL,
    homeTeam        text            NOT NULL,
    week            int                 NOT NULL,
    playerID        int                 NOT NULL REFERENCES
players(playerID),
    passAttempts int                NOT NULL DEFAULT 0,
    passCompletions int                 NOT NULL DEFAULT 0,
    passYards       int                 NOT NULL DEFAULT 0,
    passTDs             int                 NOT NULL DEFAULT 0,
    interceptions int               NOT NULL DEFAULT 0,
    timesSacked     int                 NOT NULL DEFAULT 0,
    rushAttempts int                NOT NULL DEFAULT 0,
    rushYards       int                 NOT NULL DEFAULT 0,
    rushTDs             int                 NOT NULL DEFAULT 0,
    receptions      int                 NOT NULL DEFAULT 0,
    receptionYards  int                 NOT NULL DEFAULT 0,
    receptionTDs int                NOT NULL DEFAULT 0,
FOREIGN KEY(awayTeam, homeTeam, week) REFERENCES games(awayTeam, homeTeam, week),
PRIMARY KEY(awayTeam, homeTeam, week, playerID)
);
```

Sample Data:



	awayteam	hometeam	week	playerid	passattempts	passcompletions	passyards	passtds	interceptions	timessacked	rushattempts	rushyards	rushtds	receptions	receptionyards	receptiontds
	text	text	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer	integer
1	Patriots	Jets	1	3	45	30	349	3	1	2	0	0	0	0	0	0
2	Patriots	Jets	1	7	0	0	0	0	0	0	25	120	2	3	23	0
3	Patriots	Jets	1	12	0	0	0	0	0	0	0	0	0	9	123	2
4	Patriots	Jets	1	15	0	0	0	0	0	0	0	0	0	5	65	0
5	Patriots	Jets	1	4	31	19	230	1	2	2	4	41	0	0	0	0
6	Patriots	Jets	1	6	0	0	0	0	0	0	23	98	1	2	23	0
7	Patriots	Jets	1	10	0	0	0	0	0	0	0	0	0	12	134	1
8	Patriots	Jets	1	16	0	0	0	0	0	0	0	0	0	2	16	0
9	Giants	Eagles	1	1	35	25	300	3	1	2	0	0	0	0	0	0
10	Giants	Eagles	1	5	0	0	0	0	0	0	15	101	1	3	32	0
11	Giants	Eagles	1	9	0	0	0	0	0	0	0	0	0	6	121	1
12	Giants	Eagles	1	13	0	0	0	0	0	0	0	0	0	4	45	1
13	Giants	Eagles	1	2	34	21	299	1	2	4	1	3	0	0	0	0
14	Giants	Eagles	1	8	0	0	0	0	0	0	30	143	2	7	33	0
15	Giants	Eagles	1	11	1	1	48	1	0	0	0	0	0	3	43	0
16	Giants	Eagles	1	14	0	0	0	0	0	0	0	0	0	2	30	1
17	Jets	Giants	2	4	25	12	190	0	2	7	2	-4	0	0	0	0
18	Jets	Giants	2	6	0	0	0	0	0	0	15	80	0	2	21	0
19	Jets	Giants	2	10	0	0	0	0	0	0	0	0	0	4	32	0
20	Jets	Giants	2	16	0	0	0	0	0	0	0	0	0	2	31	0
21	Jets	Giants	2	1	50	40	500	3	1	2	0	0	0	0	0	0
22	Jets	Giants	2	5	0	0	0	0	0	0	10	140	2	2	31	0
23	Jets	Giants	2	9	0	0	0	0	0	0	0	0	0	20	200	3
24	Jets	Giants	2	13	0	0	0	0	0	0	0	0	0	10	100	1
25	Eagles	Patriots	2	2	32	20	240	2	1	3	0	0	0	0	0	0
26	Eagles	Patriots	2	8	0	0	0	0	0	0	13	111	1	0	0	0
27	Eagles	Patriots	2	11	0	0	0	0	0	0	0	0	0	3	23	0
28	Eagles	Patriots	2	14	0	0	0	0	0	0	0	0	0	1	1	1
29	Eagles	Patriots	2	3	45	32	444	4	0	3	0	0	0	0	0	0
30	Eagles	Patriots	2	7	0	0	0	0	0	0	5	23	1	8	121	1
31	Eagles	Patriots	2	12	0	0	0	0	0	0	1	23	0	12	157	2
32	Eagles	Patriots	2	15	0	0	0	0	0	0	0	0	0	15	200	2

KickerStats

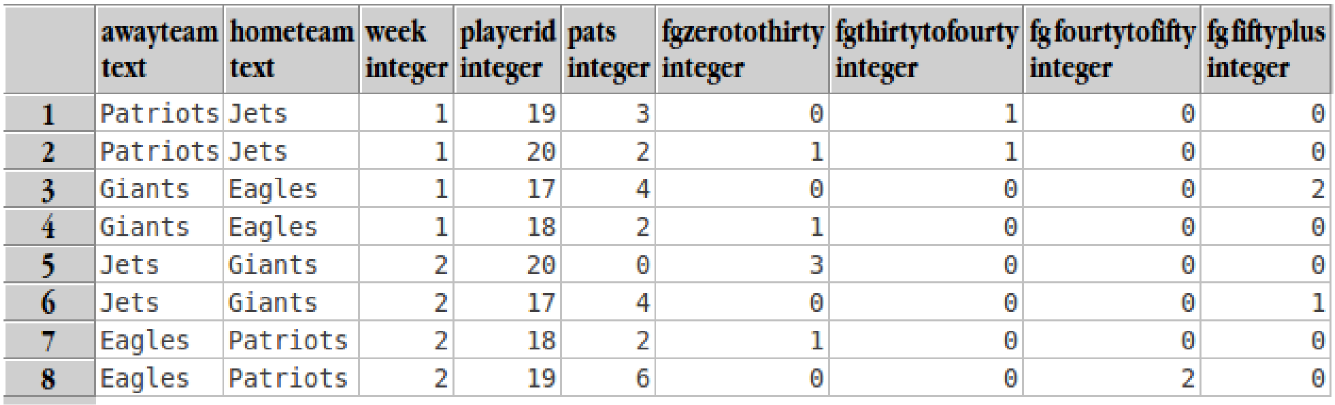
Purpose: Conatins the value of all kicker stats based on the game that was played and the player

Functional Dependencies: (awayTeam, homeTeam, week, playerID) → pats, fgZeroToThrirty, fgThirtyToFourty, fgFourtyToFifty, fgFiftyPlus

Create Statement:

```
CREATE TABLE kickerStats
(
        awayTeam            text                NOT NULL,
        homeTeam            text                NOT NULL,
        week                int                 NOT NULL,
        playerID            int                 NOT NULL REFERENCES players(playerID),
        pats                int                 NOT NULL DEFAULT 0,
        fgZeroToThirty      int                 NOT NULL DEFAULT 0,
        fgThirtyToFourty    int                 NOT NULL DEFAULT 0,
        fgFourtyToFifty     int                 NOT NULL DEFAULT 0,
        fgFiftyPlus         int                 NOT NULL DEFAULT 0,
FOREIGN KEY(awayTeam, homeTeam, week) REFERENCES games(awayTeam, homeTeam, week),
PRIMARY KEY(awayTeam, homeTeam, week, playerID)
);
```

Sample Data:



	awayteam text	hometeam text	week integer	playerid integer	pats integer	fgzerotothirty integer	fgthirtytofourty integer	fgfourtytofifty integer	fgfiftyplus integer
1	Patriots	Jets	1	19	3	0	1	0	0
2	Patriots	Jets	1	20	2	1	1	0	0
3	Giants	Eagles	1	17	4	0	0	0	2
4	Giants	Eagles	1	18	2	1	0	0	0
5	Jets	Giants	2	20	0	3	0	0	0
6	Jets	Giants	2	17	4	0	0	0	1
7	Eagles	Patriots	2	18	2	1	0	0	0
8	Eagles	Patriots	2	19	6	0	0	2	0

CurrentRosters

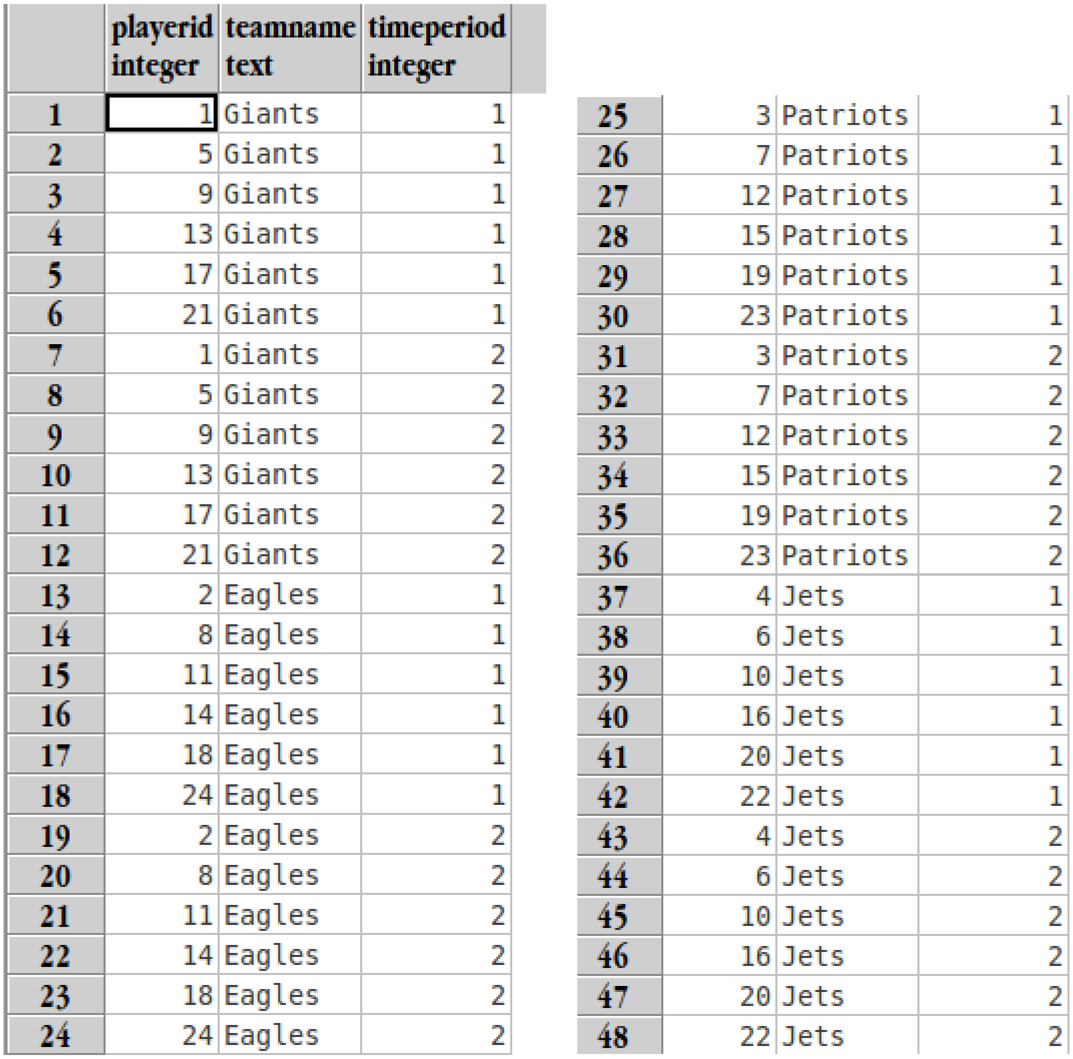
Purpose: Associates NFL teams with players and the week, so you can see who plays for what team at any given time.

Functional Dependencies: (playerID, teamName, timePeriod) →

Create Statement:

```
CREATE TABLE currentRosters
(
        playerId            int                   NOT NULL REFERENCES players(playerID),
        teamName            text                  NOT NULL REFERENCES teams(teamName),
        timePeriod          int                   NOT NULL REFERENCES
timePeriod(weekNumber),
PRIMARY KEY(playerID, teamName, timePeriod)
);
```

Sample Data:



	playerid integer	teamname text	timeperiod integer
1	1	Giants	1
2	5	Giants	1
3	9	Giants	1
4	13	Giants	1
5	17	Giants	1
6	21	Giants	1
7	1	Giants	2
8	5	Giants	2
9	9	Giants	2
10	13	Giants	2
11	17	Giants	2
12	21	Giants	2
13	2	Eagles	1
14	8	Eagles	1
15	11	Eagles	1
16	14	Eagles	1
17	18	Eagles	1
18	24	Eagles	1
19	2	Eagles	2
20	8	Eagles	2
21	11	Eagles	2
22	14	Eagles	2
23	18	Eagles	2
24	24	Eagles	2
25	3	Patriots	1
26	7	Patriots	1
27	12	Patriots	1
28	15	Patriots	1
29	19	Patriots	1
30	23	Patriots	1
31	3	Patriots	2
32	7	Patriots	2
33	12	Patriots	2
34	15	Patriots	2
35	19	Patriots	2
36	23	Patriots	2
37	4	Jets	1
38	6	Jets	1
39	10	Jets	1
40	16	Jets	1
41	20	Jets	1
42	22	Jets	1
43	4	Jets	2
44	6	Jets	2
45	10	Jets	2
46	16	Jets	2
47	20	Jets	2
48	22	Jets	2

Matchups

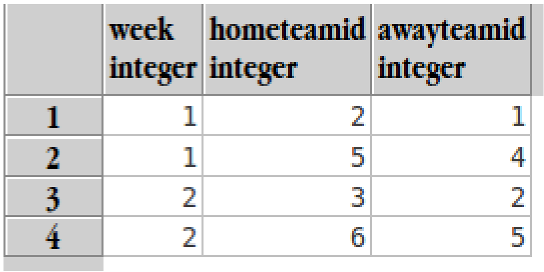
Purpose: List of matchups between fantasy teams, referencing a certain week and the two teams

Functional Dependencies: (week, homeTeamID, awayTeamID) →

Create Statement:

```
CREATE TABLE matchups
(
        week                int              NOT NULL REFERENCES timePeriod(weekNumber),
        homeTeamID          int              NOT NULL REFERENCES fantasyTeams(teamID),
        awayTeamID          int              NOT NULL REFERENCES fantasyTeams(teamID),
PRIMARY KEY(week, homeTeamID, awayTeamID)
);
```

Sample Data:



	week integer	hometeamid integer	awayteamid integer
1	1	2	1
2	1	5	4
3	2	3	2
4	2	6	5

FantasyRosters

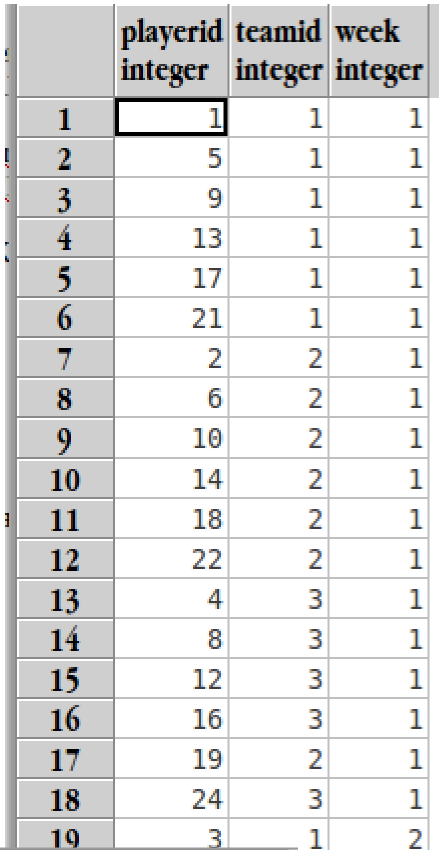
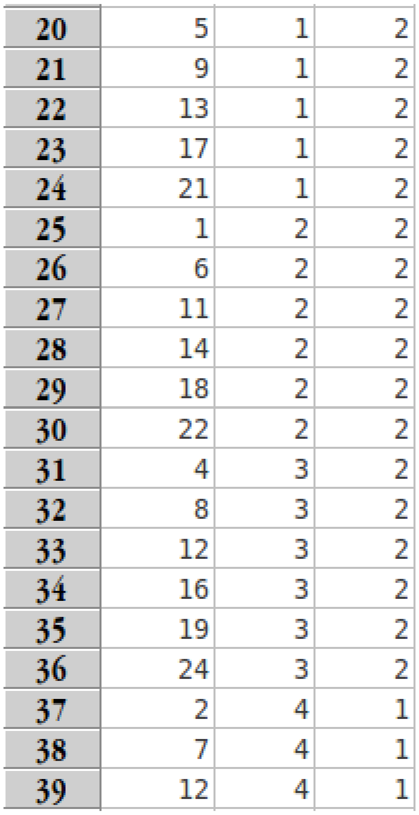
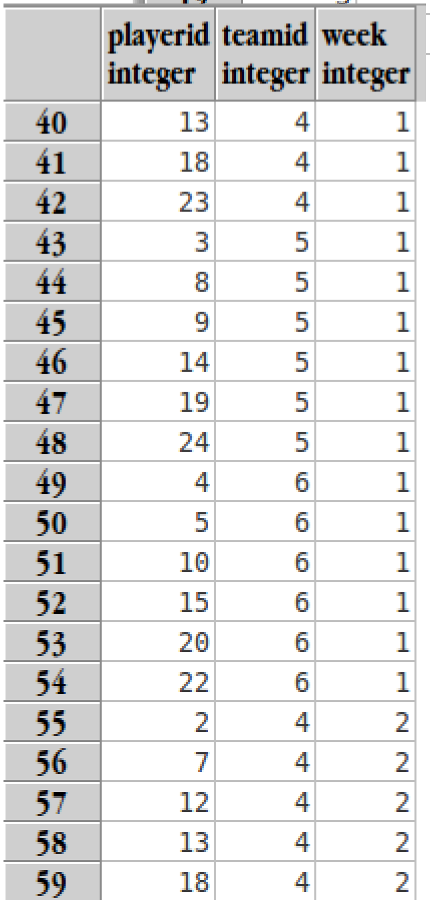
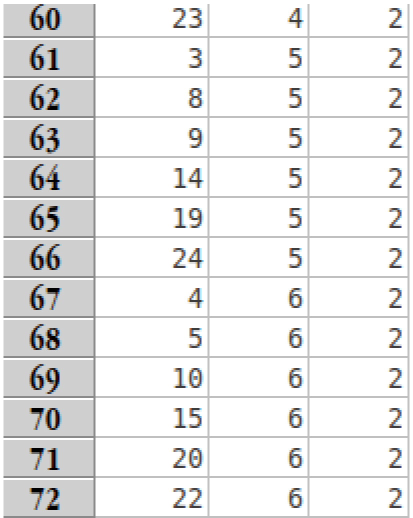
Purpose: Contains association of player to fantasy teams based on the week

Functional Dependencies: (playerID, teamID, week) →

Create Statement:

```
CREATE TABLE fantasyRosters
(
        playerID            int              NOT NULL REFERENCES players(playerID),
        teamID              int              NOT NULL REFERENCES fantasyTeams(teamID),
        week                    int             NOT NULL REFERENCES timePeriod(weekNumber),
PRIMARY KEY(playerID, teamID, week)
);
```

Sample Data:

	playerid integer	teamid integer	week integer
1	1	1	1
2	5	1	1
3	9	1	1
4	13	1	1
5	17	1	1
6	21	1	1
7	2	2	1
8	6	2	1
9	10	2	1
10	14	2	1
11	18	2	1
12	22	2	1
13	4	3	1
14	8	3	1
15	12	3	1
16	16	3	1
17	19	2	1
18	24	3	1
19	3	1	2
20	5	1	2
21	9	1	2
22	13	1	2
23	17	1	2
24	21	1	2
25	1	2	2
26	6	2	2
27	11	2	2
28	14	2	2
29	18	2	2
30	22	2	2
31	4	3	2
32	8	3	2
33	12	3	2
34	16	3	2
35	19	3	2
36	24	3	2
37	2	4	1
38	7	4	1
39	12	4	1
40	13	4	1
41	18	4	1
42	23	4	1
43	3	5	1
44	8	5	1
45	9	5	1
46	14	5	1
47	19	5	1
48	24	5	1
49	4	6	1
50	5	6	1
51	10	6	1
52	15	6	1
53	20	6	1
54	22	6	1
55	2	4	2
56	7	4	2
57	12	4	2
58	13	4	2
59	18	4	2
60	23	4	2
61	3	5	2
62	8	5	2
63	9	5	2
64	14	5	2
65	19	5	2
66	24	5	2
67	4	6	2
68	5	6	2
69	10	6	2
70	15	6	2
71	20	6	2
72	22	6	2

PointValues

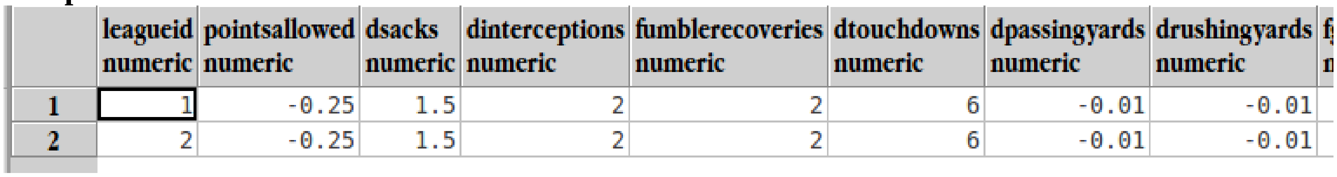
Purpose: Holds the multiplier value for each stat to determine how many point you get for each stat point.

Functional Dependencies: **l**eagueID → pointsAllowed, dSacks, dInterceptions, fumbleRecoveries, dTouchdowns, dPassingYards, dRushingYards, fgBlocked, puntBlocked, passAttempts, passCompletions, passYards, passTDs, interceptions, timesSacked, rushAttempts, rushYards, rushTDs, receptions, receptionYards, receptionTDs, pats, fgZeroToThirty, fgThirtyToFourty, fgFourtyToFifty, fgFiftyPlus

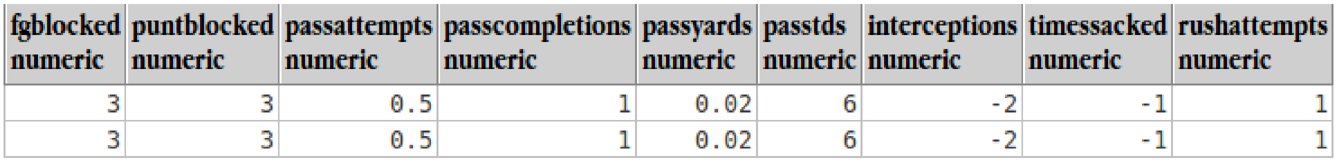
Create Statement:

```
CREATE TABLE pointValues
(
	leagueID		numeric				NOT NULL PRIMARY KEY,
	pointsAllowed		numeric				NOT NULL DEFAULT -.25,
	dSacks			numeric				NOT NULL DEFAULT 1.5,
	dInterceptions		numeric				NOT NULL DEFAULT 2,
	fumbleRecoveries	numeric				NOT NULL DEFAULT 2,
	dTouchdowns		numeric				NOT NULL DEFAULT 6,
	dPassingYards		numeric				NOT NULL DEFAULT -.01,
	dRushingYards		numeric				NOT NULL DEFAULT -.01,
	fgBlocked		numeric				NOT NULL DEFAULT 3,
	puntBlocked		numeric				NOT NULL DEFAULT 3,
	passAttempts		numeric				NOT NULL DEFAULT .5,
	passCompletions		numeric				NOT NULL DEFAULT 1,
	passYards		numeric				NOT NULL DEFAULT .02,
	passTDs			numeric				NOT NULL DEFAULT 6,
	interceptions		numeric				NOT NULL DEFAULT -2,
	timesSacked		numeric				NOT NULL DEFAULT -1,
	rushAttempts		numeric				NOT NULL DEFAULT 1,
	rushYards		numeric				NOT NULL DEFAULT .05,
	rushTDs			numeric				NOT NULL DEFAULT 6,
	receptions		numeric				NOT NULL DEFAULT 1,
	receptionYards		numeric				NOT NULL DEFAULT .05,
	receptionTDs		numeric				NOT NULL DEFAULT 6,
	pats			numeric				NOT NULL DEFAULT 1,
	fgZeroToThirty		numeric				NOT NULL DEFAULT 2,
	fgThirtyToFourty	numeric				NOT NULL DEFAULT 3,
	fgFourtyToFifty		numeric				NOT NULL DEFAULT 4,
	fgFiftyPlus		numeric				NOT NULL DEFAULT 5
);
```

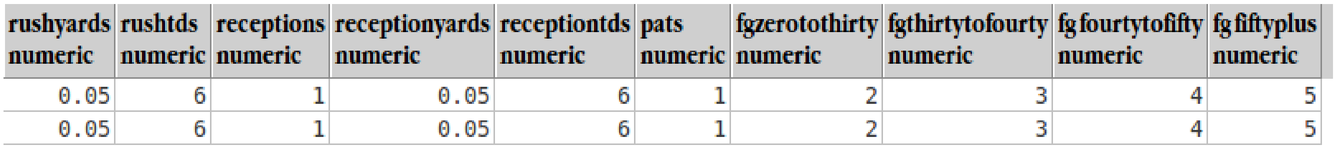
Sample Data:



	leagueid numeric	pointsallowed numeric	dsacks numeric	dinterceptions numeric	fumblerecoveries numeric	dtouchdowns numeric	dpassingyards numeric	drushingyards numeric
1	1	-0.25	1.5	2	2	6	-0.01	-0.01
2	2	-0.25	1.5	2	2	6	-0.01	-0.01



fgblocked numeric	puntblocked numeric	passattempts numeric	passcompletions numeric	passyards numeric	passtds numeric	interceptions numeric	timessacked numeric	rushattempts numeric
3	3	0.5	1	0.02	6	-2	-1	1
3	3	0.5	1	0.02	6	-2	-1	1



rushyards numeric	rushtds numeric	receptions numeric	receptionyards numeric	receptiontds numeric	pats numeric	fgzerotothirty numeric	fgthirtytofourty numeric	fgfourtytofifty numeric	fgfiftyplus numeric
0.05	6	1	0.05	6	1	2	3	4	5
0.05	6	1	0.05	6	1	2	3	4	5

Views

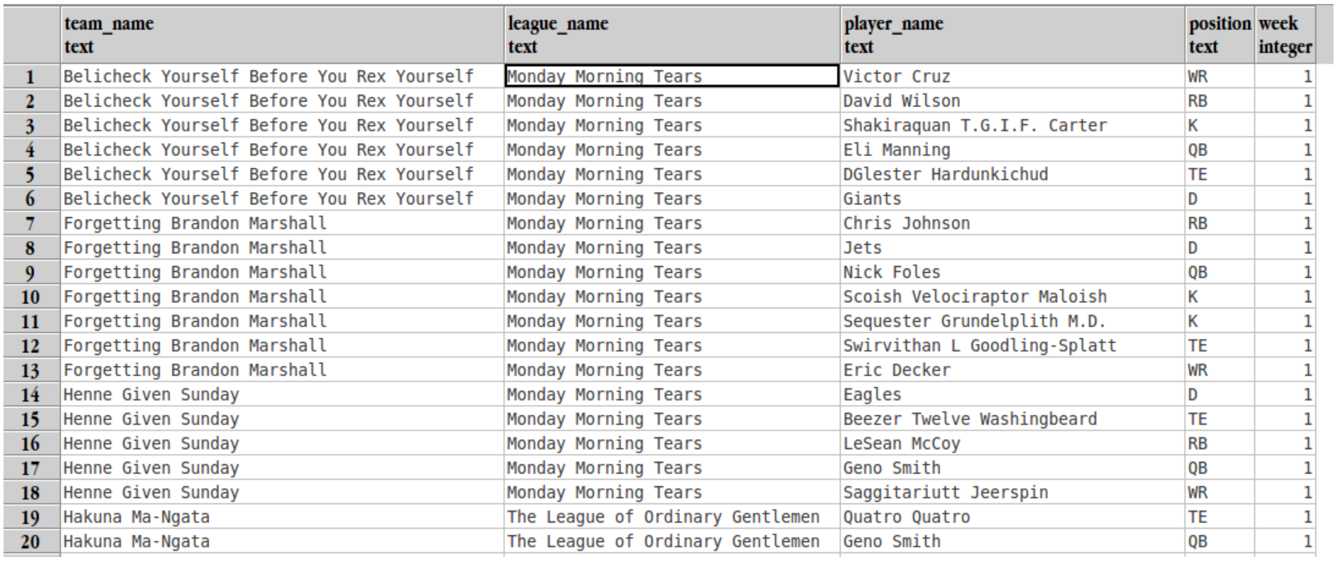
LeagueRoster

Purpose: This view allows the rosters of all teams to be seen, identified by names and not ID numbers, making it easier for users to see who is on what team and what position they play.

Create Code:

```
SELECT
                ft.teamName AS Team_Name,
                fl.leagueName AS League_Name,
                pl.name AS Player_Name,
                pl.position AS Position,
                fr.week AS Week
        FROM
                fantasyRosters fr, fantasyTeams ft, players pl, fantasyLeagues fl
        WHERE
                fr.teamID = ft.teamID
                AND fr.playerID = pl.playerID
                AND fl.leagueID = ft.leagueID
                ORDER BY fr.week, fl.leagueName, ft.teamName;
```

Sample Output:



	team_name text	league_name text	player_name text	position text	week integer
1	Belicheck Yourself Before You Rex Yourself	Monday Morning Tears	Victor Cruz	WR	1
2	Belicheck Yourself Before You Rex Yourself	Monday Morning Tears	David Wilson	RB	1
3	Belicheck Yourself Before You Rex Yourself	Monday Morning Tears	Shakiraquan T.G.I.F. Carter	K	1
4	Belicheck Yourself Before You Rex Yourself	Monday Morning Tears	Eli Manning	QB	1
5	Belicheck Yourself Before You Rex Yourself	Monday Morning Tears	DGlester Hardunkichud	TE	1
6	Belicheck Yourself Before You Rex Yourself	Monday Morning Tears	Giants	D	1
7	Forgetting Brandon Marshall	Monday Morning Tears	Chris Johnson	RB	1
8	Forgetting Brandon Marshall	Monday Morning Tears	Jets	D	1
9	Forgetting Brandon Marshall	Monday Morning Tears	Nick Foles	QB	1
10	Forgetting Brandon Marshall	Monday Morning Tears	Scoish Velociraptor Maloish	K	1
11	Forgetting Brandon Marshall	Monday Morning Tears	Sequester Grundelplith M.D.	K	1
12	Forgetting Brandon Marshall	Monday Morning Tears	Swirvithan L Goodling-Splatt	TE	1
13	Forgetting Brandon Marshall	Monday Morning Tears	Eric Decker	WR	1
14	Henne Given Sunday	Monday Morning Tears	Eagles	D	1
15	Henne Given Sunday	Monday Morning Tears	Beezer Twelve Washingbeard	TE	1
16	Henne Given Sunday	Monday Morning Tears	LeSean McCoy	RB	1
17	Henne Given Sunday	Monday Morning Tears	Geno Smith	QB	1
18	Henne Given Sunday	Monday Morning Tears	Saggitariutt Jeerspin	WR	1
19	Hakuna Ma-Ngata	The League of Ordinary Gentlemen	Quatro Quatro	TE	1
20	Hakuna Ma-Ngata	The League of Ordinary Gentlemen	Geno Smith	QB	1

Reports

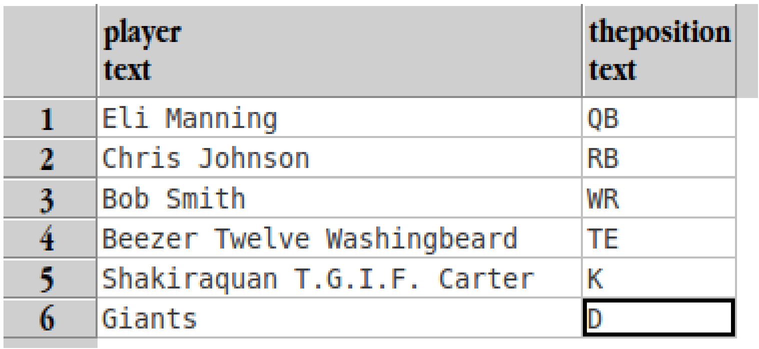
GetFreeAgents

Purpose: This tells the user what players are free agents for a certain week, so they know who they can add to their team.

Function call:

```
SELECT *
        FROM getFreeAgents(leagueID, weekNumber);
```

Sample Output:



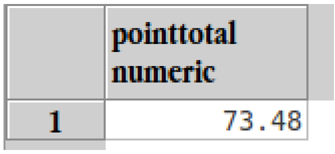
	player	theposition
	text	text
1	Eli Manning	QB
2	Chris Johnson	RB
3	Bob Smith	WR
4	Beezer Twelve Washingbeard	TE
5	Shakiraquan T.G.I.F. Carter	K
6	Giants	D

PointsTotal

Purpose: Return the amount of points a player had during a week, which is calculated from the stats multiplied by the point values

Function Call: select pointTotal(playerID, weekNumber, leagueID);

Sample output: select pointTotal(3, 1, 1)



	pointtotal numeric
1	73.48

Stored Procedures

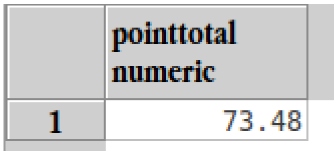
pointTotal()

Purpose: This function returns the total number of fantasy points a player scores in a given week for a leagues point scheme

Create Statement:

```
CREATE OR REPLACE FUNCTION pointTotal(thePlayerID int, theWeek int, theLeagueID int)
RETURNS numeric as $$
        DECLARE
                total numeric;
        BEGIN
                IF (SELECT position FROM players WHERE players.playerID = thePlayerID) = 'D'
                THEN
                        total := (((SELECT pv.pointsAllowed
                        FROM pointValues pv
                        WHERE pv.leagueID = theLeagueID) *
                        (SELECT ds.pointsAllowed
                        FROM defenseStats ds
                        WHERE ds.week = theWeek AND ds.playerID = thePlayerID)) +

                        ((SELECT pv.dSacks
                        FROM pointValues pv
                        WHERE pv.leagueID = theLeagueID) *
                        (SELECT ds.sacks
                        FROM defenseStats ds
                        WHERE ds.week = theWeek AND ds.playerID = thePlayerID)) +

                        ((SELECT pv.dInterceptions
                        FROM pointValues pv
                        WHERE pv.leagueID = theLeagueID) *
                        (SELECT ds.interceptions
                        FROM defenseStats ds
                        WHERE ds.week = theWeek AND ds.playerID = thePlayerID)) +

                        ((SELECT pv.fumbleRecoveries
                        FROM pointValues pv
                        WHERE pv.leagueID = theLeagueID) *
                        (SELECT        ds.fumbleRecoveries
                        FROM defenseStats ds
                        WHERE ds.week = theWeek AND ds.playerID = thePlayerID)) +
                        ((SELECT pv.dTouchdowns
                        FROM pointValues pv
                        WHERE pv.leagueID = theLeagueID) *
                        (SELECT ds.touchdowns
                        FROM defenseStats ds
                        WHERE ds.week = theWeek AND ds.playerID = thePlayerID)) +

                        ((SELECT pv.dPassingYards
                        FROM pointValues pv
                        WHERE pv.leagueID = theLeagueID) *
                        (SELECT ds.passingYards
                        FROM defenseStats ds
                        WHERE ds.week = theWeek AND ds.playerID = thePlayerID)) +

                        ((SELECT pv.dRushingYards
                        FROM pointValues pv
                        WHERE pv.leagueID = theLeagueID) *
                        (SELECT ds.rushingYards
                        FROM defenseStats ds
                        WHERE ds.week = theWeek AND ds.playerID = thePlayerID)) +

                        ((SELECT pv.fgBlocked
                        FROM pointValues pv
                        WHERE pv.leagueID = theLeagueID) *
                        (SELECT ds.fgBlocked
                        FROM defenseStats ds
                        WHERE ds.week = theWeek AND ds.playerID = thePlayerID)) +

                        ((SELECT pv.puntBlocked
                        FROM pointValues pv
                        WHERE pv.leagueID = theLeagueID) *
                        (SELECT ds.puntBlocked
                        FROM defenseStats ds
                        WHERE ds.week = theWeek AND ds.playerID = thePlayerID)));

                ELSEIF (SELECT position FROM players WHERE players.playerID = thePlayerID) =
'K'
                THEN
                        total := (((SELECT pv.pats
                        FROM pointValues pv
                        WHERE pv.leagueID = theLeagueID) *
                        (SELECT ks.pats
                        FROM kickerStats ks
                        WHERE ks.week = theWeek AND ks.playerID = thePlayerID)) +

                        ((SELECT pv.fgZeroToThirty
                        FROM pointValues pv
                WHERE pv.leagueID = theLeagueID) *
                (SELECT ks.fgZeroToThirty
                FROM kickerStats ks
                WHERE ks.week = theWeek AND ks.playerID = thePlayerID)) +

                ((SELECT pv.fgThirtyToFourty
                FROM pointValues pv
                WHERE pv.leagueID = theLeagueID) *
                (SELECT ks.fgThirtyToFourty
                FROM kickerStats ks
                WHERE ks.week = theWeek AND ks.playerID = thePlayerID)) +

                ((SELECT pv.fgFourtyToFifty
                FROM pointValues pv
                WHERE pv.leagueID = theLeagueID) *
                (SELECT ks.fgFourtyToFifty
                FROM kickerStats ks
                WHERE ks.week = theWeek AND ks.playerID = thePlayerID)) +

                ((SELECT pv.fgFiftyPlus
                FROM pointValues pv
                WHERE pv.leagueID = theLeagueID) *
                (SELECT ks.fgFiftyPlus
                FROM kickerStats ks
                WHERE ks.week = theWeek AND ks.playerID = thePlayerID)));

            ELSE
                total := (((SELECT pv.passAttempts
                FROM pointValues pv
                WHERE pv.leagueID = theLeagueID) *
                (SELECT os.passAttempts
                FROM offenseStats os
                WHERE os.week = theWeek AND os.playerID = thePlayerID)) +

                ((SELECT pv.passCompletions
                FROM pointValues pv
                WHERE pv.leagueID = theLeagueID) *
                (SELECT os.passCompletions
                FROM offenseStats os
                WHERE os.week = theWeek AND os.playerID = thePlayerID)) +

                ((SELECT pv.passYards
                FROM pointValues pv
                WHERE pv.leagueID = theLeagueID) *
                (SELECT os.passYards
                FROM offenseStats os
                WHERE os.week = theWeek AND os.playerID = thePlayerID)) +
((SELECT pv.passTDs
FROM pointValues pv
WHERE pv.leagueID = theLeagueID) *
(SELECT os.passTDs
FROM offenseStats os
WHERE os.week = theWeek AND os.playerID = thePlayerID)) +

((SELECT pv.interceptions
FROM pointValues pv
WHERE pv.leagueID = theLeagueID) *
(SELECT os.interceptions
FROM offenseStats os
WHERE os.week = theWeek AND os.playerID = thePlayerID)) +

((SELECT pv.timesSacked
FROM pointValues pv
WHERE pv.leagueID = theLeagueID) *
(SELECT os.timesSacked
FROM offenseStats os
WHERE os.week = theWeek AND os.playerID = thePlayerID)) +

((SELECT pv.rushAttempts
FROM pointValues pv
WHERE pv.leagueID = theLeagueID) *
(SELECT os.rushAttempts
FROM offenseStats os
WHERE os.week = theWeek AND os.playerID = thePlayerID)) +

((SELECT pv.rushYards
FROM pointValues pv
WHERE pv.leagueID = theLeagueID) *
(SELECT os.rushYards
FROM offenseStats os
WHERE os.week = theWeek AND os.playerID = thePlayerID)) +

((SELECT pv.rushTDs
FROM pointValues pv
WHERE pv.leagueID = theLeagueID) *
(SELECT os.rushTDs
FROM offenseStats os
WHERE os.week = theWeek AND os.playerID = thePlayerID)) +

((SELECT pv.receptions
FROM pointValues pv
WHERE pv.leagueID = theLeagueID) *
(SELECT os.receptions
            FROM offenseStats os
            WHERE os.week = theWeek AND os.playerID = thePlayerID)) +

            ((SELECT pv.receptionYards
            FROM pointValues pv
            WHERE pv.leagueID = theLeagueID) *
            (SELECT os.receptionYards
            FROM offenseStats os
            WHERE os.week = theWeek AND os.playerID = thePlayerID)) +

            ((SELECT pv.receptionTDs
            FROM pointValues pv
            WHERE pv.leagueID = theLeagueID) *
            (SELECT os.receptionTDs
            FROM offenseStats os
            WHERE os.week = theWeek AND os.playerID = thePlayerID)));
        END IF;

        RETURN total;
    END;
$$ LANGUAGE plpgsql;
```

Sample Output: select pointTotal(3, 1, 1)



	pointtotal numeric
1	73.48

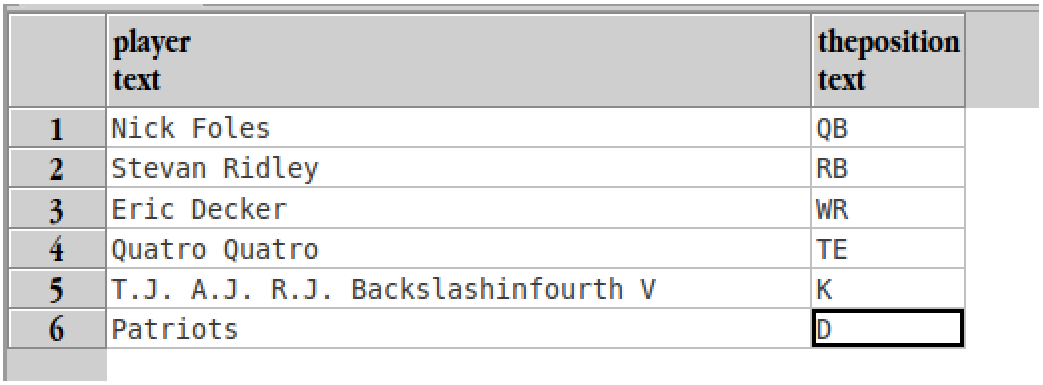
getFreeAgents()

Purpose: This function returns free agents in a table when you give it the parameters of leagueID and weekNumber

Code Create:

```
CREATE OR REPLACE FUNCTION getFreeAgents(theLeagueID INTEGER, theWeek INTEGER)
RETURNS TABLE (player text, thePosition text) AS $$
    DECLARE
        previous_week INTEGER;
    BEGIN

    IF
        theWeek = 1
    THEN
        previous_week := 1;
    ELSE
        previous_week := theWeek -1;
    END IF;

    RETURN QUERY
    SELECT
            pl.name AS playerName,
            pl.position AS positionName

    FROM
            players pl LEFT OUTER JOIN (SELECT * FROM (fantasyRosters fr JOIN
fantasyTeams ft ON fr.teamID =  ft.teamID) WHERE ft.leagueID = theLeagueID AND fr.week =
theWeek) ftr ON (pl.playerID = ftr.playerID)

    WHERE
            ftr.playerID IS NULL;

    END;
$$ LANGUAGE plpgsql;
```

Sample Output: SELECT * FROM getFreeAgents(1, 2)



	player text	theposition text
1	Nick Foles	QB
2	Stevan Ridley	RB
3	Eric Decker	WR
4	Quatro Quatro	TE
5	T.J. A.J. R.J. Backslashinfourth V	K
6	Patriots	D

playerConflict()

Purpose: Checks to see if a player is on another team in the league when a user tries to add a new player to his team via a trigger. If the player trying to be added is already on another team in the league, an error is thrown and the player is not added

Create Code:

```
CREATE OR REPLACE FUNCTION playerConflict() RETURNS TRIGGER AS $$
        DECLARE
                team_league INTEGER := (SELECT          ft.leagueID
                                        FROM fantasyTeams ft, fantasyRosters fr
                                        WHERE ft.teamID = fr.teamID
                                                AND ft.teamID = NEW.teamID
                                        LIMIT 1);
        BEGIN
                IF
                        EXISTS (SELECT fr.teamID
                                        FROM fantasyRosters fr, fantasyTeams ft, timePeriod tp
                                        WHERE fr.teamID = ft.teamID
                                                AND NEW.playerID = fr.playerID
                                                AND team_league = ft.leagueID
                                                AND tp.weekNumber = fr.week
                                                AND NEW.week = fr.week)
                                LIMIT 1
                        THEN
                                RAISE EXCEPTION 'Player is on another team in the league';
                        END IF;
                        RETURN NEW;
        END;
$$ LANGUAGE plpgsql;
```

Triggers

check_conflict

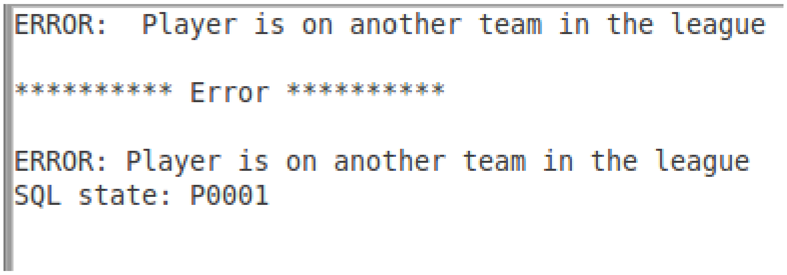
Purpose: This trigger checks before an update on fantasyRosters to see if that player already belongs to a team in that league, because a player is only allowed to be on one team in a league.

Code:

```
CREATE TRIGGER check_conflict BEFORE INSERT ON fantasyRosters
        FOR EACH ROW EXECUTE PROCEDURE playerConflict();
```

Sample Data:

```
INSERT INTO fantasyRosters (playerID, teamID, week) VALUES

        (1, 1, 1),
        (5, 1, 1),
        (9, 1, 1),
        (13, 1, 1),
        (17, 1, 1),
        (13,2,1)
```



```
ERROR:  Player is on another team in the league

********** Error **********

ERROR: Player is on another team in the league
SQL state: P0001
```

An error was correctly thrown here because player 13 was trying to be added to team 2 for week one, when he is already on team1. Because team 1 and team 3 are in the same league, it is not allowed.

Security

Security for the database is necessary, because people would try and cheat if they were able to access and modify anything they wanted. Therefore 3 roles were created, There is the average client, the commissioner of the league and the database administrator.

```
CREATE ROLE dbAdmin
GRANT SELECT, INSERT, UPDATE ON
ALL TABLES IN SCHEMA public TO dbAdmin;


CREATE ROLE regularUser
GRANT SELECT, INSERT
ON fantasyTeams, fantasyRosters
TO regularUser;

CREATE ROLE commissioner
GRANT SELECT, INSERT, UPDATE
ON fantasyTeams, fantasyRosters, fantasyLeagues, pointValues
TO commissioner;
```

Implementation Notes

Known Problems

There are some problems with the database at the current time. There is no way to account for players who can play multiple positions. Right now the position is limited to 1 at a time, but that could be expanded in the future. There are also issues in dealing with the logic of the table because of the limit of players in league. There is no good way to express that in terms of built in rules when creating the table, so that has to be worked around. There is also a limited number of stats being kept right now, for the sake of sanity, but there are potentially more stats that people could award points for,

Future Enhancements

There is a good amount of space for this to grow, on both the fantasy football side and the stat keeping one. I would like to make it make it so the user can have bench players and starting players, instead of just a roster. There also certain rules, like waiver wires that are specific to fantasy football that are not implemented in the database. They would give pick up priority to certain teams based on their past waiver activity, There are also certain tasks like trading players that could be added. Fantasy football has grown pretty robust and customizable recently, and this database doesn't have the level of flexibility that you would see from ESPN, or Yahoo fantasy football.