NFL DataBase

Created By:Matt Byam

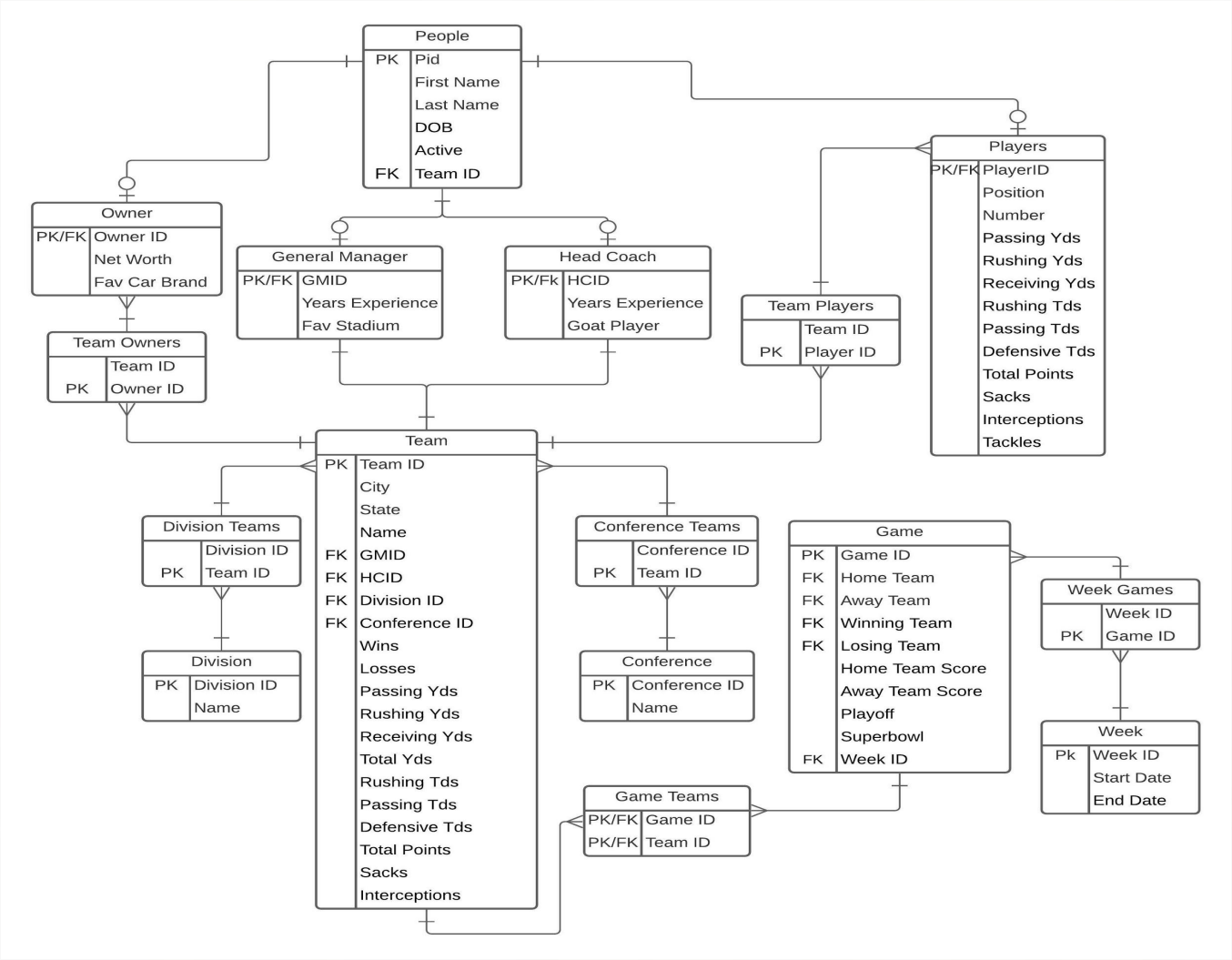
Table of Contents

Executive summary

This Document is detailing an NFL database that is comprised of a fraction of the normal data that would be stored in a full league database. This database is limited to the 2019 season, with eight teams, four teams from each conference and two teams from each division. This database contains statistics for players and teams that are factual with the exception of a few minor adjustments. The only fictional data is the Owners Favorite Car Brand, General Managers years experience and Favorite Stadium, Head Coach's years experience and GOAT player preference, and finally the sample data where Alan Labouseur is substituted for Tom Brady.

The rest of this presentation will display an ER diagram, all the tables created, stored procedures, triggers, views, interesting AND painful queries, security, and finally conclude with notes, problems, and enhancements for future development of this database.

ER Diagram


People
PK Pid
First Name
Last Name
DOB
Active
FK Team ID
Owner
PK/FK Owner ID
Net Worth
Fav Car Brand
General Manager
PK/FK GMID
Years Experience
Fav Stadium
Head Coach
PK/Fk HCID
Years Experience
Goat Player
Players
PK/FK PlayerID
Position
Number
Passing Yds
Rushing Yds
Receiving Yds
Rushing Tds
Passing Tds
Defensive Tds
Total Points
Sacks
Interceptions
Tackles
Team Owners
Team ID
PK Owner ID
Team Players
Team ID
PK Player ID
Team
PK Team ID
City
State
Name
FK GMID
FK HCID
FK Division ID
FK Conference ID
Wins
Losses
Passing Yds
Rushing Yds
Receiving Yds
Total Yds
Rushing Tds
Passing Tds
Defensive Tds
Total Points
Sacks
Interceptions
Division Teams
Division ID
PK Team ID
Division
PK Division ID
Name
Conference Teams
Conference ID
PK Team ID
Conference
PK Conference ID
Name
Game
PK Game ID
FK Home Team
FK Away Team
FK Winning Team
FK Losing Team
Home Team Score
Away Team Score
Playoff
Superbowl
FK Week ID
Week Games
Week ID
PK Game ID
Week
Pk Week ID
Start Date
End Date
Game Teams
PK/FK Game ID
PK/FK Team ID

Create Statements

&

Sample Data

People Table

The People table is the parent table for all the subtypes such as Owners, General Managers, Head Coaches, and Players. The data within the table is First Name, Last Name, Date of Birth, Activity, and Team ID based on Pid.

Pid → First Name, Last Name, DOB, Active, TeamID

```
create type Activity as ENUM ('Active', 'Inactive');
Create table People(
            Pid Char(8) not null unique,
            First_Name text not null,
            Last_Name text not null,
            DOB date not null,
            Active Activity,
            Team ID Char(8) not null,
      primary key (Pid)
);
```

	pid [PK] character (8)	first_name text	last_name text	dob date	active activity	teamid character (8)
1	o1	Robert	Kraft	1941-06...	Active	t1
2	o2	Woody	Johnson	1947-04...	Active	t2
3	o3	Lamar	Hunt	1932-08...	Active	t3
4	o4	Pat	Bowlen	1944-02...	Active	t4
5	o5	John	Elway	1960-06...	Inactive	t4
6	o6	Sheila	Hamp	1951-05...	Active	t5
7	o7	Virginia Halas	McCaskey	1923-01...	Active	t6
8	o8	Gayle	Benson	1947-01...	Active	t7
9	o9	David	Tepper	1957-09...	Active	t8
10	gm1	Joe	Douglas	1976-05...	Active	t2
11	gm2	Brett	Veach	1977-12...	Active	t3
12	gm3	Bob	Quinn	1976-07...	Active	t5
13	gm4	Ryan	Pace	1977-02...	Active	t6
14	gm5	Mickey	Loomis	1956-08...	Active	t7
15	gm6	Marty	Hurney	1949-12...	Active	t8
16	hc1	Bill	Belichick	1952-03...	Active	t1
17	hc2	Adam	Gase	1978-03...	Active	t2
18	hc3	Andy	Reid	1958-03...	Active	t3
19	hc4	Vance	Joseph	1972-09...	Active	t4
20	hc5	Matt	Patricia	1970-01...	Active	t5

Players Table

The information in this table is everything about the players in the league such as Position, Number, Passing Yards, Rushing Yards, Receiving Yards, Total yards, Rushing TDs, Passing TDs, Receiving TDs, Defensive TDs, Total Points, Sacks, Interceptions, and Tackles all based on Player ID.

Player ID → Position, Number, Passing Yds, Rushing Yds, Receiving Yds, Total Yds, Rushing TDs, Passing TDs, Receiving TDs, Defensive TDs, Total Points, Sacks, Interceptions, Tackles

```
Create table Players(
        Playerid Char(8) not null unique,
        Position text not null,
        Number int,
        Passing_Yds int not null,
        Rushing_Yds int not null,
        Receiving_Yds int not null,
        Total_Yds int not null,
        Rushing_TDs int not null,
        Passing_TDs int not null,
        receiving_TDs int not null,
        Defensive_TDs int not null,
        Total_Points int not null,
        Sacks real not null,
        Interceptions int  not null,
        Tackles int not null,
    primary key (PlayerID)
);
```

	playerid [PK] character (8)	position text	number integer	passing_yds integer	rushing_yds integer	receiving_yds integer	total_yds integer	rushing_tds integer	passing_tds integer	receiving_tds integer	defensive_tds integer	total_points integer	sacks real	interceptions integer	tackles integer
1	p1	QB	12	4057	34	0	4091	3	24	0	0	162	0	0	0
2	p2	CB	24	0	0	0	0	0	0	0	2	12	0	6	44
3	p3	QB	14	3024	62	0	3062	2	19	0	0	126	0	0	0
4	p4	RB	26	0	789	461	1250	3	1	0	0	24	0	0	0
5	p5	QB	15	4031	218	0	4249	2	26	0	0	168	0	0	0
6	p6	WR	10	0	23	860	883	0	0	7	0	42	0	0	0

General Manager Table

```
Create table GeneralManager(
          GMID Char(8) not null unique,
          Years_Exp real not null,
          Fav_Stadium text,
     primary key (GMID)
);
```

Table containing information on all General Managers. Holds the General managers Years Experience and Favorite Stadium based on the General Manager ID.

GMID → Years Exp, Fav Stadium

	gmid [PK] character (8)	years_exp real	fav_stadium text
1	hc1	30	Gillette Stadium
2	gm1	12	Soldier Field
3	gm2	9	Ford Field
4	o5	25	Lambeau field
5	gm3	14	Heinz Field
6	gm4	35	Superdom
7	gm5	4	Heinz Field
8	gm6	5	Lambeau Field

Head Coach Table

Table containing all the Head Coaches information. This includes Years Experience and GOAT Player based on HCID (Head Coach ID).

HCID → Years Exp, GOAT Player

```
Create table HeadCoach(
          HCID Char(8) not null unique,
          Years_Exp real not null,
          GOAT_Player text,
     primary key (HCID)
);
```

	hcid [PK] character (8)	years_exp real	goat_player text
1	hc1	30	Jerry Rice
2	hc2	14	Tom Brady
3	hc3	24	Tom Brady
4	hc4	19	Joe Montana
5	hc5	4	Lawrence Taylor
6	hc6	25	Tom Brady
7	hc7	7	Jerry Rice
8	hc8	9	Reggie White

Owners Table

Table containing all Owners information. This includes Net Worth and Favorite Car Brand based on the Owner ID.

	ownerid [PK] character (8)	net_worth bigint	fav_carbrand text
1	o1	6600000000	Bently
2	o2	4200000000	Mercedes
3	o3	15300000000	Ferarri
4	o4	1000000000	BMW
5	o5	145000000	Bently
6	o6	2500000000	BMW
7	07	1300000000	Ford
8	o8	3300000000	Rolls Royce
9	o9	13000000000	Lamborghini

```
Create table Owners(
        OwnerID Char(8) not null unique,
        Net_Worth bigint not null,
        Fav_CarBrand text,
    primary key (OwnerID)
);
```

Owner ID → Net Worth, Fav Car Brand

Team Table

The Team table includes the information about all the teams within the scope of the database. This information includes City, State, Team Name, General Manager ID, Head Coach ID, Division ID, Conference ID, Wins, Losses, Passing Yards, Rushing Yards, Total Yards, Rushing TDs, Passing TDs, Defensive TDs, Total Points, Sacks, and Interceptions all based on the Team ID.

Team ID → City, State, Team Name, GMID, HCID, Division ID, Conference ID, Wins, Losses, Passing Yds, Rushing Yds, Total Yds, Rushing TDs, Passing TDs, Defensive TDs, Total Points, Sacks, Interceptions

```
Create Table Team(
        TeamID Char(8) not null unique,
        City text not null,
        State text not null,
        Team_Name text not null,
        GMID Char(8) not null,
        HCID Char(8) not null,
        DivisionID Char(8) not null,
        Conference Char(8) not null,
        Wins int not null,
        Losses int not null,
        Passing_Yds int not null,
        Rushing_Yds int not null,
        Total_Yds int not null,
        Rushing_TDs int not null,
        Passing_TDs int not null,
        Defensive_TDs int not null,
        Total_Points int not null,
        Sacks int not null,
        Interceptions int  not null,
    primary key(TeamID)
);
```

	teamid [PK] character (8)	city text	state text	team_name text	gmid character (8)	hcid character (8)	divisionid character (8)	conferenceid character (8)	wins integer	losses integer	passing_yds integer	rushing_yds integer	total_yds integer	rushing_tds integer	passing_tds integer	defensive_tds integer	total_points integer	sacks integ
1	t1	Boston	Massac...	Patriots	hc1	hc1	d1	c1	12	4	3961	1703	5664	17	25	5	420	
2	t2	New Y...	New York	Jets	hc2	gm1	d1	c1	7	9	3111	1257	4368	6	19	5	276	
3	t3	Kansa...	Missouri	Chiefs	hc3	gm2	d2	c1	12	4	4498	1569	6067	16	30	3	451	
4	t4	Denver	Colarado	Broncos	hc4	o5	d2	c1	7	9	3115	1662	4777	11	16	1	282	

Conference Table

```
Create table Conference(
        ConferenceID Char(8) not null unique,
        Name text not null,
    primary key (ConferenceID)
);
```

The table includes all the information about the Conferences. This information the name based on the Conference ID.

Conference ID → Name

	conferenceid [PK] character (8)	name text
1	c1	AFC
2	c2	NFC

Division Table

```
Create table Division(
        DivisionID Char(8) not null unique,
        Name text not null,
    primary key (DivisionID)
);
```

The table includes all the information about the Divisions. This information the name based on the Division ID.

Division ID → Name

	divisionid [PK] character (8)	name text
1	d1	AFC East
2	d2	AFC West
3	d3	AFC North
4	d4	AFC South
5	d5	NFC East
6	d6	NFC West
7	d7	NFC North
8	d8	NFC South

Week Table

The table includes all the information about the Weeks during the NFL season. The information contained is the Start Date and End Date based on the Week ID.

Week ID → Start Date, End Date


```
Create table Week(
        WeekID Char(8) not null unique,
        Start_Date date not null,
        End_Date date not null,
    primary key (WeekID)
);
```

	weekid [PK] character (8)	start_date date	end_date date
1	w1	2019-09-09	2019-09-16
2	w2	2019-09-17	2019-09-23
3	w3	2019-09-24	2019-09-30
4	w4	2019-10-01	2019-10-07
5	w5	2019-10-08	2019-10-14
6	w6	2019-10-15	2019-10-21
7	w7	2019-10-22	2019-10-28
8	w8	2019-10-29	2019-11-04
9	w9	2019-11-05	2019-11-11
10	w10	2019-11-12	2019-11-18
11	w11	2019-11-19	2019-11-25

Game Table

The table includes all the information about each Game. This information includes Home Team, Away Team, Winning Team, Losing Team, Home Team Score, Losing Team Score, Playoff Game, SuperBowl, And WeekID all based on the Game ID.

Game ID → Home Team, Away Team, Winning Team, Losing Team, Home Team Score, Away Team Score, Playoff Game, SuperBowl, Week ID

```
create type YN as ENUM ('Yes', 'No');
Create table Game(
        GameID Char(8) not null unique,
        Home_Team char(8) not null,
        Away_Team char(8) not null,
        Winning_Team Char(8) not null,
        Loosing_Team Char(8) not null,
        Home_Team_Score int not null,
        Away_Team_Score int not null,
        Playoff_Game YN not null,
        SuperBowl YN not null,
        WeekID Char(8) not null,
    primary key (GameID)
);
```

	gameid [PK] character (8)	home_team character (8)	away_team character (8)	winning_team character (8)	loosing_team character (8)	home_team_score integer	away_team_score integer	playoff_game yn	superbowl yn	weekid character (8)
1	g1	t4	t6	t6	t4	14	16	No	No	w2
2	g2	t1	t2	t1	t2	30	14	No	No	w3
3	g3	t5	t3	t3	t5	30	34	No	No	w4
4	g4	t4	t3	t3	t4	6	30	No	No	w7
5	g5	t6	t7	t7	t6	25	36	No	No	w7
6	g6	t2	t1	t1	t2	0	33	No	No	w7
7	g7	t6	t5	t6	t5	20	13	No	No	w10

Division Teams Table

```
Create Table DivisionTeams(
          DivisionID Char(8) not null,
          TeamID Char(8) not null unique,
     primary key(TeamID)
);
```

The table includes all the teams in each division. The information held is the Division ID which is based on the Team ID.

Team ID → Division ID

	divisionid character (8)	teamid [PK] character (8)
1	d1	t1
2	d1	t2
3	d2	t3
4	d2	t4
5	d7	t5
6	d7	t6
7	d8	t7
8	d8	t8

Game Teams Table

```
Create Table GameTeams(
          GameID Char(8) not null,
          TeamID Char(8) not null,
     primary key(GameID, TeamID)
);
```

The table includes all games for each team. The information in the table is the Game ID and Team ID which combine to form a composite primary key.

GameID, TeamID → GameID, TeamID

	gameid [PK] character (8)	teamid [PK] character (8)
1	g1	t4
2	g1	t6
3	g2	t1
4	g2	t2
5	g3	t5
6	g3	t4
7	g4	t4
8	g4	t3
9	g5	t6
10	g5	t7
11	g6	t2
12	g6	t1
13	g7	t6
14	g7	t5

Conference Teams Table

```
Create Table ConferenceTeams(
        ConferenceID Char(8) not null,
        TeamID Char(8) not null unique,
    primary key(TeamID)
);
```

The table includes all the teams that are in which conference. The information within this table is Conference ID based on the Team ID.

Team ID → Conference ID

	conferenceid character (8)	teamid [PK] character (8)
1	c1	t1
2	c1	t2
3	c1	t3
4	c1	t4
5	c2	t5
6	c2	t6
7	c2	t7
8	c2	t8

Week Games Table

```
Create Table WeekGames(
        GameID Char(8) not null unique
        WeekID Char(8) not null,
    primary key(GameID)
);
```

The table includes all the games in each week. The information within this table is Week ID based on Game ID.

Game ID → Week ID

	weekid character (8)	gameid [PK] character (8)
1	w2	g1
2	w3	g2
3	w4	g3
4	w7	g4
5	w7	g5
6	w7	g6
7	w10	g7
8	w12	g8
9	w13	g9
10	w14	g10
11	w15	g11
12	w16	g12
13	w16	g13

Team Owners Table

```
Create Table TeamOwners(
          TeamID Char(8) not null,
          OwnerID Char(8) not null unique,
     primary key(OwnerID)
);
```

The table includes all the Owners of each team. The information included is the Team ID which is based on the Owner ID.

Owner ID → Team ID

	teamid character (8)	ownerid [PK] character (8)
1	t1	o1
2	t2	o2
3	t3	o3
4	t4	o4
5	t4	p5
6	t5	p6
7	t6	o7
8	t7	o8
9	t8	o9

Team Players Table

The table includes all the players on each team . The information within this table is Team ID based on Player ID.

Player ID → Team ID

```
Create Table TeamPlayers(
        TeamID Char(8) not null,
        PlayerID Char(8) not null unique,
    primary key(PlayerID)
);
```

	teamid character (8)	playerid [PK] character (8)
1	t1	p1
2	t1	p2
3	t2	p3
4	t2	p4
5	t3	p5
6	t3	p6
7	t4	p7
8	t4	p8
9	t5	p9
10	t5	p10
11	t6	p11
12	t6	p12
13	t7	p13

Stored Procedures

Triggers

Views

Stored Procedures

This stored Procedure allows users to quickly access the front office (General Manager and Head Coach) of any team within the database, by entering the team ID.

The example table on the side show the output when ‘t2’ or the Broncos are entered into this function.

```
CREATE OR REPLACE FUNCTION get_frontoffice_by_team(char(8), refcursor) returns REFCURSOR
AS $$
DECLARE
  team char(8) := $1;
  resultSet REFCURSOR := $2;
BEGIN
  OPEN resultset for
        select p.pid,p.first_name, p.last_name
              from people p
                    inner join generalmanager gm on p.pid = gm.gmid
              where p.teamID = team
                    union
              select p.pid,p.first_name,p.last_name
              from people p
                    inner join headcoach hc on p.pid = hc.hcid
              where p.teamid = team;
  return resultset;
END;$$
LANGUAGE PLPGSQL;

select get_frontoffice_by_team ('t2','results');
Fetch all from results;
```

	pid character (8)	first_name text	last_name text
1	o5	John	Elway
2	hc4	Vance	Joseph

Triggers

The purpose of this trigger is that when a new team is inserted into the database, the name will be referenced across the rest of the league to see if that name is already being used, and if so then the insert will be rejected.

```
create or replace function Team_Validation()
returns trigger as
$$
begin
      if (new.Team_name = team.team_name) then
      delete from team where team_name = new.team_name;
      end if;
      return new;
      end; $$
      language plpgsql;

create trigger Team_Validation
after insert on Team
for each row
execute procedure Team_Validation();
```

```
insert into team (TeamID,city,state,team_name,HCID,GMID,divisionID,conferenceID,wins,losses,passing_yds,Rushing_yds,
                  Total_yds,rushing_tds,passing_tds,defensive_tds,total_points,sacks,interceptions) values
        ('t9','Las Vegas','Nevada', 'Saints','hc4','gm2','d1','c1',12,4,3961,1703,5664,17,25,5,420,47,25);
```

```
QUERY:  SELECT (new.Team_name = team.team_name)
CONTEXT:  PL/pgSQL function team_validation() line 3 at IF
SQL state: 42P01
```

League Top 5 Passing Leaders View

Some of the most beloved views in football are seeing the top statistics leaders in the league in their respective categories.

This view is showing the Top 5 players with the most passing yards within the scope of this database.

```
Create view passing_leaders as
select p.first_name,p.last_name,pl.passing_yds
from people p
        inner join players pl on p.pid = pl.playerId
order by pl.passing_yds desc
limit 5;
```

	first_name text	last_name text	passing_yds integer
1	Alan	Labouseur	4057
2	Patrick	Mahomes	4031
3	Sam	Darnold	3024
4	Drew	Breese	2979
5	Teddy	Bridgwater	1384

League Top 5 Rushing Leaders View

Continuing off the last view, it is necessary to add a top 5 leading rusher view, which displays the top 5 players with the most rushing yards within the scope of the database.

```
create view rushing_leaders as
select p.first_name,p.last_name,pl.rushing_yds
from people p
        inner join players pl on p.pid = pl.playerId
order by pl.rushing_yds desc
limit 5;
```

	first_name text	last_name text	rushing_yds integer
1	Christian	McCaffrey	1387
2	LeVeon	Bell	789
3	Patrick	Mahomes	218
4	Tarik	Cohen	213
5	Sam	Darnold	62

Interesting & Painful Queries

Interesting Query 1

The first interesting query is one that returns the playerID, first name, and last name of a player who was on a winning team in week 7, and plays Quarterback as their position.

The Table at the bottom shows the output for this query and can see that the three quarterbacks that won in week 7 were Alan Labouseur (player 1), Teddy Bridgewater (player 14), and Patrick Mahomes (player 5).

```
select distinct pl.playerID, p.first_name,p.last_name
from people p
     inner join players pl on p.pid = pl.playerID
     inner join teamplayers tp on pl.playerID =
tp.playerId
     inner join team t on tp.teamID = t.teamID
     inner join gameteams gt on t.teamId = gt.teamID
     inner join game g on gt.gameID = g.gameID
where t.teamID = g.winning_team
and g.weekID = 'w7'
and pl.position ='QB';
```

	playerid character (8)	first_name text	last_name text
1	p1	Alan	Labouseur
2	p14	Teddy	Bridgwater
3	p5	Patrick	Mahomes

Interesting Query 2

The second interesting query returns the First Name, Last Name, DOB, Team Name, and Division ID of a Head Coach and GM of a team that won an away game. Also, the General Manager was born in 1977 and the Head Coach had to be born in 1978.

The Table below show the output of the chiefs GM and the entire bears front office fit this criteria.

```
select distinct p.first_name, p.last_name, p.DOB, t.Team_Name, dt.DivisionID
from people p
      inner join generalmanager gm on p.pid = gm.gmid
      inner join team t on p.teamID = t.teamID
      inner join divisionteams dt on t.teamID = dt.teamID
      inner join gameteams gt on t.teamID = gt.teamID
      inner join game g on gt.teamID = g.winning_team
where p.teamID = g.away_team
and p.DOB >='01/01/1977'
and p.DOB<= '12/31/1977'
      union
select distinct p.first_name,p.last_name,p.DOB, t.Team_Name, dt.DivisionID
from people p
      inner join headcoach hc on p.pid = hc.hcid
      inner join team t on p.teamID = t.teamID
      inner join divisionteams dt on t.teamID = dt.teamID
      inner join gameteams gt on t.teamID = gt.teamID
      inner join game g on gt.teamID = g.winning_team
where p.teamID = g.away_team
and p.DOB >='01/01/1978'
and p.DOB<= '12/31/1978'
```

	first_name text	last_name text	dob date	team_name text	divisionid character (8)
1	Brett	Veach	1977-12...	Chiefs	d2
2	Matt	Nagy	1978-04...	Bears	d7
3	Ryan	Pace	1977-02...	Bears	d7

Security, Roles & More

Security & Roles

Commissioner: Has the power to change anything within the database, and acts as the admin role for the league.

```
Create role commissioner;
Grant all on all tables in schema public to commissioner;
```

Front Office: This is made up of the General Manager and the Head Coach and they have control over the players table and are able to move players in and out of teams.

```
Create role front_office;
Grant update on players to front_office;
Grant insert on players to front_office;
Grant select on all tables in schema public to
front_office;
```

Player: Only has the ability to view the database.

```
Create role player;
Grant select on all tables in schema public to
player;
```

Owner: Has the power to change anything within the player or the team tables.

```
Create role owner;
Grant insert on headcoach, generalmanager, players
to owner;
Grant update on headcoach, generalmanager, players
to owner;
Grant select on all tables in schema public to
owner;
```

This database contains four roles which are Commissioner, Front Office, Owner, and Player, each with their own abilities depending on how much control they have over the properties included in the database

Notes - Future Enhancements - Known Problems

- This database is only a fraction of the size that it could possibly be, only including 8 of the 32 teams in the NFL.
- Adding a Season Table would allow this database to become more multidimensional by encompassing more than one season.
- More input validations like Team_Validation() can be added to secure the data integrity from possible errors being added to the database.
- There has to be a more efficient way of organizing the people subtypes, so when you want to include more than one but less than all four groups you don't need to use the union function. Future update of this database should include this with more effort focused on this issue.
- Create a separate table with player stats and team stats so that it is not necessary to add all the stats when creating one of those two entities.
- Envisioned this database to encompass an entire league with hundreds of records of players and stats but would be possible with future enhancements and work.