Milestone #2: Mid Project

NFL Play statistics Dataset

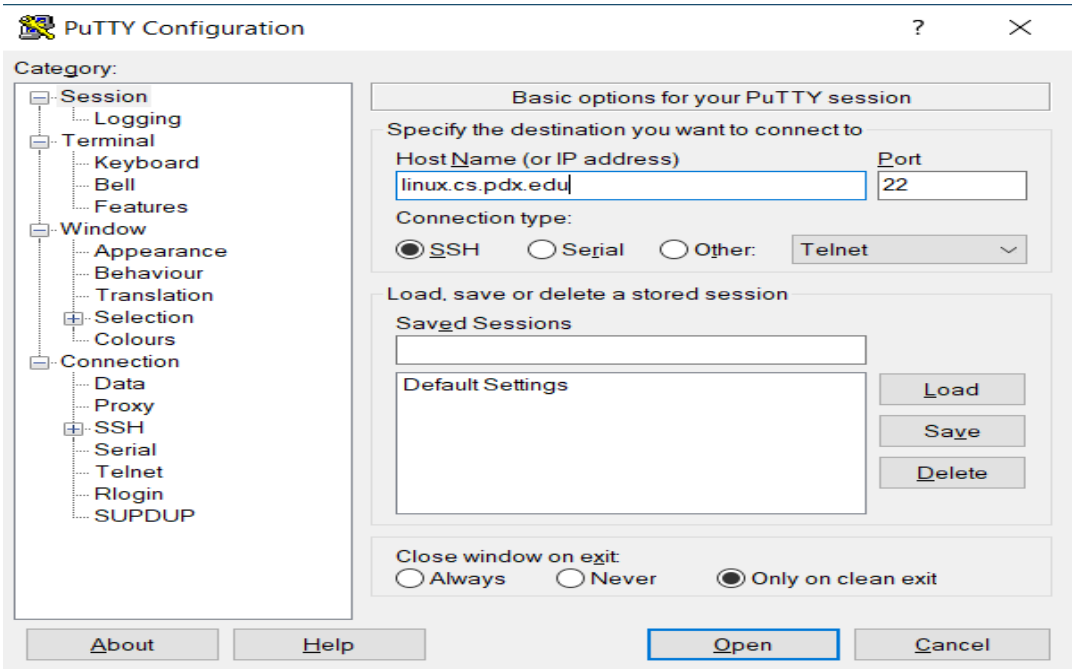
Submitted by - ***Chitradevi Maruthavanan***

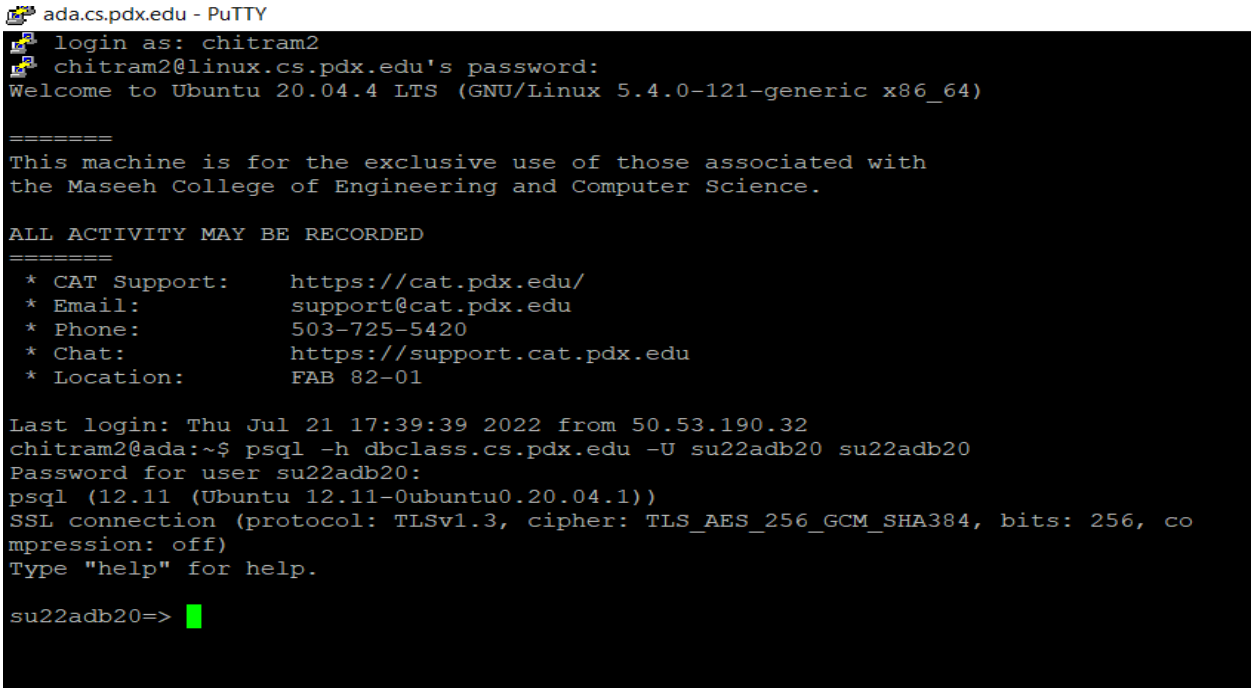
Reasons to change the proposal from milestone #1:

The COVID Vaccine dataset at https://github.com/nychealth/covid-vaccine-data had issues in relating one table with other. I could not establish any foreign key for the tables. So, I chose the new dataset on NFL Play statistics from https://www.kaggle.com/datasets/toddsteussie/nfl-play-statistics-dataset-2004-to-present?resource=download

Postgres installation Steps:

To show that I have successfully installed Postgres on my virtual machine. The screenshot is below:





Source of data:

I am going to use the NFL (National Football League) Play statistics dataset. This NFL dataset provides play-by-play data from the 2004 to 2019 seasons. The dataset is at https://www.kaggle.com/datasets/toddsteussie/nfl-play-statistics-dataset-2004-to-present?resource=download

The dataset contains the following CSV files which contains a lot of data as elaborated below

1. plays.csv – Contains important data related to the game such as play type, possession team, non-possession team, field position, net yards etc.
2. games.csv – Contains game related data such as game time, season, weeks etc
3. kicks.csv – Contains kicks related data for the games such as Kick type, Kick outcome etc
4. interceptions.csv – Contains data related to interception position, interception yards etc
5. fumbles.csv – Contains information such as fumble type, fumble position and fumble turn around.
6. tackles.csv – Contains data such as tackle type, tackle position and tackle yards
7. gameParticipation.csv- Contains information such as game Participant name, college detail, participant home city details

These CSV files are transferred to the pdx linux machines using pscp as below.

pscp D:\DB_Project_files chitram2@linux.cs.pdx.edu:/u/chitram2/dbproject*

```
PS C:\Users\kshya> pscp D:\DB_Project_files\* chitram2@linux.cs.pdx.edu:/u/chitram2/dbproject/
chitram2@linux.cs.pdx.edu's password:
fumbles.csv               | 727 kB | 727.6 kB/s | ETA: 00:00:00 | 100%
gameParticipation.csv     | 20384 kB | 20384.4 kB/s | ETA: 00:00:00 | 100%
games.csv                 | 445 kB | 445.1 kB/s | ETA: 00:00:00 | 100%
interceptions.csv         | 210 kB | 210.0 kB/s | ETA: 00:00:00 | 100%
kicks.csv                 | 10142 kB | 10142.2 kB/s | ETA: 00:00:00 | 100%
plays.csv                 | 101228 kB | 1177.1 kB/s | ETA: 00:00:00 | 100%
tackles.csv               | 21225 kB | 21225.6 kB/s | ETA: 00:00:00 | 100%
```

Next step is to create and populate the tables in the database using below steps.

The **data types** used in my tables are

1. Text
2. Int
3. Float
4. Date

Table Creation:

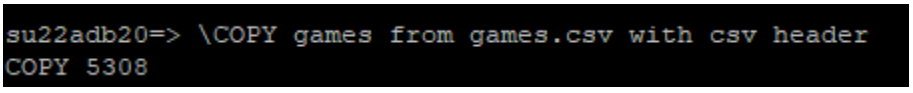
1. **Games** table

   Data Preprocessing:
   I use the data as it is. No Preprocessing was done.

   CREATE Command for table
   CREATE TABLE games(gameId int,season int,week int,gameDate date,gameTimeEastern text,gameTimeLocal text,homeTeamId int,visitorTeamId int,seasonType text,weekNameAbbr text,siteId int,homeTeamDistance int,visitingTeamDistance int, homeTeamFinalScore int,visitingTeamFinalScore int,winningTeam int);

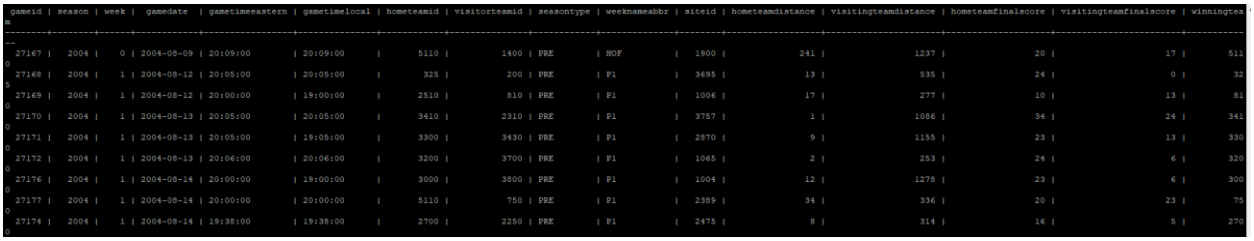
   Copy data from CSV to table:

   \COPY games from games.csv with csv header



Screenshot of the populated table:

SELECT * FROM games;



gameid	season	week	gamedate	gametimeeastern	gametimelocal	hometeamid	visitorteamid	seasontype	weeknameabbr	siteid	hometeamdistance	visitingteamdistance	hometeamfinalscore	visitingteamfinalscore	winningteam
27167	2004	0	2004-08-09	20:09:00	20:09:00	5110	1400	PRE	HOF	1900	241	1237	20	17	5110
27168	2004	1	2004-08-12	20:05:00	20:05:00	325	200	PRE	P1	3695	13	535	24	0	325
27169	2004	1	2004-08-12	20:00:00	19:00:00	2510	810	PRE	P1	1006	17	277	10	13	810
27170	2004	1	2004-08-13	20:05:00	20:05:00	3410	2310	PRE	P1	3757	1	1086	34	24	3410
27171	2004	1	2004-08-13	20:05:00	19:05:00	3300	3430	PRE	P1	2870	9	1155	23	13	3300
27172	2004	1	2004-08-13	20:06:00	20:06:00	3200	3700	PRE	P1	1065	2	253	24	6	3200
27176	2004	1	2004-08-14	20:00:00	19:00:00	3000	3800	PRE	P1	1004	12	1278	23	6	3000
27177	2004	1	2004-08-14	20:00:00	20:00:00	5110	750	PRE	P1	2389	34	336	20	23	5110
27174	2004	1	2004-08-14	19:38:00	19:38:00	2700	2250	PRE	P1	2475	8	314	16	5	2700

Cardinality of the table:

SELECT COUNT(*) FROM games;

```
su22adb20=> SELECT COUNT(*) FROM games;
 count
-------
  5308
(1 row)
```

Primary key:

ALTER TABLE games add constraint pkey_games primary key(gameid);

```
su22adb20=> ALTER TABLE games add constraint pkey_games primary key(gameid);
ALTER TABLE
```

2. **Plays** table

Data Preprocessing:

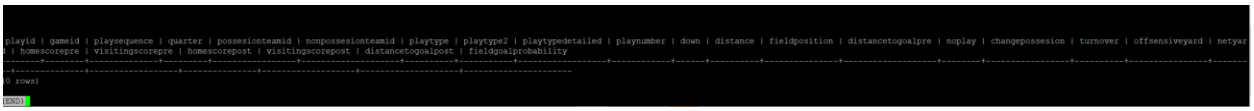
I removed few columns – gameclock, safety, firstdown, efficientplay, evpre, evpost, evplay, downconversion, huddle and formation because data is not sufficiently populated.

CREATE Command for table

CREATE TABLE plays(playid int, gameid int, playsequence int, quarter int, possesionteamid int, nonpossesionteamid int, playtype text, playtype2 text, playtypedetailed text, playnumber int ,down int, distance int, fieldposition text, distancetogoalpre int, noplay int, playDescription text, playStats text, playDescriptionFull text, changepossesion int, turnover int, offsensiveyard int, net yard int, homescorepre int, visitingscorepre int ,homescorepost int, visitingscorepost int, distanceToGoalPost text, fieldGoalProbabi lity text);

```
su22adb20=> CREATE TABLE plays(playid int,gameid int,playsequence int,quarter int,possesionteamid int,nonpossesionteamid int,pl
aytype text,playtype2 text,playtypedetailed text,playnumber int,down int,distance int,fieldposition text,distancetogoalpre int,

noplay int,playDescription text,playStats text,playDescriptionFull text,changepossesion int,turnover int,offsensiveyard int,net
yard int,homescorepre int,visitingscorepre int,homescorepost int,visitingscorepost int,distanceToGoalPost text,fieldGoalProbabi
lity text);
CREATE TABLE
su22adb20=> SELECT * FROM plays
```



playid | gameid | playsequence | quarter | possesionteamid | nonpossesionteamid | playtype | playtype2 | playtypedetailed | playnumber | down | distance | fieldposition | distancetogoalpre | noplay | changepossesion | turnover | offsensiveyard | netyar
d | homescorepre | visitingscorepre | homescorepost | visitingscorepost | distancetogoalpost | fieldgoalprobability

(0 rows)

(END)

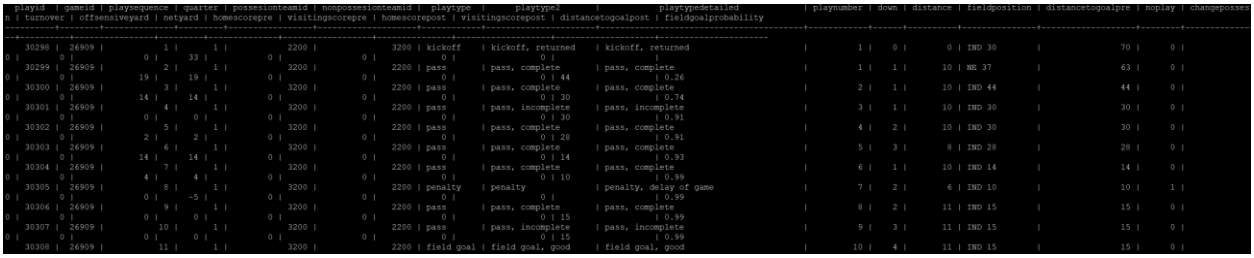
Copy data from CSV to table:

\COPY plays from plays.csv with csv header

```
su22adb20=> \COPY plays from plays.csv with csv header
COPY 870384
```

Screenshot of the populated table:

SELECT * FROM plays;



Cardinality of the table:

SELECT COUNT(*) FROM plays;

```
su22adb20=> SELECT COUNT (*) FROM plays
su22adb20-> ;
 count
--------
 870384
(1 row)
```

Primary key:

ALTER TABLE plays add constraint pkey_plays primary key(playId);

```
su22adb20=> ALTER TABLE plays add constraint pkey_plays primary key(playId);
ALTER TABLE
```

Foreign key:

ALTER TABLE plays add constraint fk_gameid foreign key(gameid) references games(gameid);

```
su22adb20=> ALTER TABLE plays add constraint fk_gameid foreign key(gameid) references games(gameid);
ALTER TABLE
```

3. **Fumbles** table

Data Preprocessing:

I use the data as it is. No Preprocessing was done.

CREATE Command for table

CREATE TABLE **fumbles** (fumid int, playid int, teamid int, playerid int, fumposition text, fumtype text, fumoob int, fumturnover float, fumnull int);

```
su22adb20=> CREATE TABLE fumbles (fumid int,playid int,teamid int,playerid int,fumposition text,fumtype text,fumoob int,fumturn
over int,fumnull int);
CREATE TABLE
su22adb20=> SELECT * FROM fumbles
su22adb20-> ;
 fumid | playid | teamid | playerid | fumposition | fumtype | fumoob | fumturnover | fumnull
-------+--------+--------+----------+-------------+---------+--------+-------------+---------
(0 rows)
```

Copy data from CSV to table:

\COPY fumbles from fumbles.csv with csv header

```
su22adb20=> \COPY fumbles from fumbles.csv with csv header
COPY 14910
```

Screenshot of the populated table:

SELECT * FROM fumbles;

fumid	playid	teamid	playerid	fumposition	fumtype	fumoob	fumturnover	fumnull
800001	1	3800	20020185	RB	forced	0	0	0
800002	27	3200	20000239	FB	forced	0	0	0
800003	53	3200	20000199	QB	unforced	0	1	0
800004	56	3800	20020081	QB	forced	0	0	0
800005	84	3800	19970057	LB	forced	1	0	0
800006	174	3300	19990131	QB	unforced	0	1	0
800007	182	3300	19990131	QB	forced	0	0	0
800008	188	3300	19990438	RB	unforced	0	1	0
800009	222	3800	19900017	RB	forced	0	1	0
800010	333	3800	20020081	QB	forced	0	0	0
800011	466	3800	19900017	RB	forced	0	0	0
800012	498	3800	19960417	WR/KR	unforced	0	0	0
800013	522	3430	20010016	WR/KR	unforced	0	0	0
800014	578	3430	20010053	QB	forced	0	0	0
800015	612	3430	19950074	RB	forced	0	0	0
800016	615	3800	20030054	WR	forced	0	1	0
800017	644	3800	20020081	QB	forced	0	0	0

Cardinality of the table:

SELECT COUNT(*) FROM fumbles;

```
su22adb20=> SELECT COUNT (*) FROM fumbles
su22adb20-> ;
 count
-------
 14910
(1 row)
```

Primary Key:

ALTER TABLE fumbles add constraint pkey_fumbles primary key(fumid);

```
su22adb20=> ALTER TABLE fumbles add constraint pkey_fumbles primary key(fumid);
ALTER TABLE
```

Foreign key:

ALTER TABLE fumbles add constraint fk_plays foreign key(playid) references plays(playid);

```
su22adb20=> ALTER TABLE fumbles add constraint fk_plays foreign key(playid) references plays(playid);
ALTER TABLE
```

4. **Kicks** table

Data Preprocessing:

I use the data as it is. No Preprocessing was done.

CREATE Command for table

CREATE TABLE kicks (kickid int, playid int, teamid int, playerid int, kickposition text, kicktype text, kickoutcome text, kickinside20 int, kickonside int, kickOwnRecovery int, kickLength int, kickReturnYds int, kickNetYds int, kickReturnTd int, kicknull int);

Copy data from CSV to table:

\COPY kicks from kicks.csv with csv header

```
su22adb20=> \COPY kicks from kicks.csv with csv header
COPY 148014
```

Screenshot of the populated table:

SELECT * FROM kicks;

kickid	playid	teamid	playerid	kickposition	kicktype	kickoutcome	kickinside20	kickonside	kickownrecovery	kicklength	kickreturnyds	kicknetyds	kickreturntd	kicknull
300001	1	3200	19960452	K	kickoff	returned	1	0	0	72.0	17	55.0	0	0
300002	7	3800	19940992	P	punt	returned	1	0	0	41.0	5	36.0	0	0
300003	15	3200	19940450	P	punt	touchback	0	0	0	53.0	0	53.0	0	0
300004	20	3800	19940992	P	punt	returned	0	0	0	55.0	22	33.0	0	0
300005	31	3200	19960452	K	xp	good	0	0	0	19.0	0	0.0	0	0
300006	32	3200	19960452	K	kickoff	returned	1	0	0	73.0	18	55.0	0	0
300007	43	3200	19960452	K	xp	good	0	0	0	19.0	0	0.0	0	0
300008	44	3200	19960452	K	kickoff	returned	0	0	0	70.0	28	42.0	0	0
300009	48	3800	19940992	P	punt	returned	0	0	0	43.0	-2	45.0	0	0
300010	57	3800	20000169	K	field goal	good	0	0	0	51.0	0	0.0	0	0
300011	58	3800	20000169	K	kickoff	returned	0	0	0	70.0	23	47.0	0	0

Cardinality of the table:

SELECT COUNT(*) FROM kicks;

```
su22adb20=> SELECT COUNT(*) FROM kicks;
 count
--------
 148014
(1 row)
```

Primary Key:

ALTER TABLE kicks add constraint pkey_kicks primary key(kickid);

```
su22adb20=> ALTER TABLE kicks add constraint pkey_kicks primary key(kickid);
ALTER TABLE
```

Foreign key:

ALTER TABLE kicks add constraint fk_players foreign key(playid) references plays(playid);

```
su22adb20=> ALTER TABLE kicks add constraint fk_playid foreign key(playid) references plays(playid);
ALTER TABLE
```

5. **Interception** table

Data Preprocessing:

Used pandas library in python to remove duplicate rows in the table.

Code:

```
>>>import pandas as pd
>>> file_name = r"D:\DB_Project_files\interceptions_original.csv"
>>> file_name_output = r"D:\DB_Project_files\interceptions.csv"
>>> df = pd.read_csv(file_name)
>>> df.drop_duplicates(subset=None, inplace=True)
>>> df.to_csv(file_name_output, index=False)
```

CREATE Command for table

CREATE TABLE **interceptions** (interceptionid int, playid int, teamid int, playerid int, intposition text, int int, intyards int, intTd int, intNull int);

```
u22adb20=> CREATE TABLE interceptions(interceptionid int,playid int,teamid int,playerid int,intposition text,int int,intyards int,intTd int,intNull int);
REATE TABLE
u22adb20=> SELECT  * FROM interceptions;
interceptionid | playid | teamid | playerid | intposition | int | intyards | inttd | intnull
---------------+--------+--------+----------+-------------+-----+----------+-------+---------
0 rows)
```

Copy data from CSV to table:

\COPY interceptions from interceptions.csv with csv header

```
su22adb20=> \COPY interceptions from interceptions.csv with csv header
COPY 9463
```

Screenshot of the populated table:

SELECT * FROM interception;

interceptionid	playid	teamid	playerid	intposition	int	intyards	inttd	intnull
540001	36	3200	20030036	DB	1	14	0	0
540002	60	3800	20000091	CB	1	0	0	0
540003	84	3800	19970057	LB	1	65	0	0
540004	130	3200	20030036	DB	1	13	0	0
540005	602	3430	20040012	LB	1	6	0	0
540006	632	3430	20000102	CB	1	14	0	0
540007	637	3430	19920005	CB	1	18	0	0
540008	838	3800	20000091	CB	1	0	0	0
540009	915	3800	20010482	S	1	1	0	0
540010	942	3800	20010202	CB	1	2	0	0
540011	1002	3800	20040033	OLB	1	2	0	0

Cardinality of the table:

SELECT COUNT(*) FROM interception;

```
u22adb20=> SELECT COUNT(*) FROM interceptions;
count
------
 9463
1 row)
```

Primary key:

ALTER TABLE interceptions add constraint pkey_interceptions primary key(interceptionid);

```
su22adb20=>  ALTER TABLE interceptions add constraint pkey_interceptions primary key(interceptionid);
ALTER TABLE
```

Foreign key:

ALTER TABLE interceptions add constraint fk_playid foreign key(playid) references plays(playid);

```
su22adb20=> ALTER TABLE interceptions add constraint fk_playid foreign key(playid) references plays(playid);
ALTER TABLE
```

6. **Tackles** table

Data Preprocessing:

Used pandas library in python to remove duplicate rows in the table.

Code:

```
>>>import pandas as pd
>>> file_name = r"D:\DB_Project_files\tackles_original.csv"
>>> file_name_output = r"D:\DB_Project_files\tackles.csv
>>> df = pd.read_csv(file_name)
>>> df.drop_duplicates(subset=None, inplace=True)
>>> df.to_csv(file_name_output, index=False)
```

CREATE Command for table

CREATE TABLE **tackles** (tackleid int, playid int, teamid int, playerid int, tackleposition text, tackletype text, tackleYdsScrim text, scrim text, tacklenull int);

```
u22adb20=> CREATE TABLE tackles(tackleid int,playid int,teamid int,playerid int,tackleposition text,tackletype text,tackleYdsScrim text,scrim text,tacklenull int);
REATE TABLE
u22adb20=> SELECT * FROM tackles;
tackleid | playid | teamid | playerid | tackleposition | tackletype | tackleydsscrim | scrim | tacklenull
---------+--------+--------+----------+----------------+------------+----------------+-------+------------
0 rows)
```

Copy data from CSV to table:

\COPY tackles from tackles.csv with csv header

```
su22adb20=> \COPY tackles from tackles.csv with csv header
COPY 726294
```

Screenshot of the populated table:

SELECT * FROM tackles;

tackleid	playid	teamid	playerid	tackleposition	tackletype	tackleydsscrim	scrim	tacklenull
555001	1	3200	19950022	CB	solo		in bounds	0
555002	2	3200	19950022	CB	solo	8.0	in bounds	0
555003	3	3200	19950023	CB	solo	8.0	in bounds	0
555004	4	3200	20030036	DB	solo	8.0	in bounds	0
555005	6	3200	19940145	S	solo	1.0	in bounds	0
555006	7	3800	20010054	CB	solo		in bounds	0
555007	8	3800	20020619	S	solo	17.0	in bounds	0
555008	9	3800	19990048	DT	solo	8.0	in bounds	0
555009	11	3800	20010493	LB	solo	5.0	in bounds	0
555010	16	3200	19950023	CB	solo	14.0	in bounds	0
555011	17	3200	19970091	LB	solo	-2.0	in bounds	0
555012	19	3200	19970091	LB	solo	3.0	in bounds	0
555013	20	3800	20010459	CB	solo		in bounds	0
555014	21	3800	19970086	DE	solo	5.0	in bounds	0
555015	23	3800	19970183	CB	solo	4.0	in bounds	0
555016	24	3800	20010064	S	solo	-1.0	in bounds	0
555017	25	3800	19970183	CB	solo	7.0	in bounds	0

Cardinality of the table:

SELECT COUNT(*) FROM tackles;

```
u22adb20=> SELECT COUNT(*) FROM tackles;
count
-------
726294
1 row)
```

Primary Key:

ALTER TABLE tackles add constraint pkey_tackles primary key(tackleid);

```
su22adb20=> ALTER TABLE tackles add constraint pkey_tackles primary key(tackleid);
ALTER TABLE
```

7. Game Participation table

Data Preprocessing:

Used pandas library in python to remove duplicate rows in the table.

Code:

```
>>>import pandas as pd
>>> file_name = r"D:\DB_Project_files\gameParticipation_original.csv"
>>> file_name_output = r"D:\DB_Project_files\gameParticipation.csv
>>> df = pd.read_csv(file_name)
>>> df.drop_duplicates(subset=None, inplace=True)
>>> df.to_csv(file_name_output, index=False)
```

CREATE Command for table

CREATE TABLE gameParticipation(gamePartId text,gameId int,teamId text,playerId text,gamePartUnit text,gamePartSnapCount text,nameFirst text,nameLast text,nameFull text,position text,college text,heightInches text,weight text,dob date,ageAtDraft float,homeCity text,homeState text,homeCountry text);

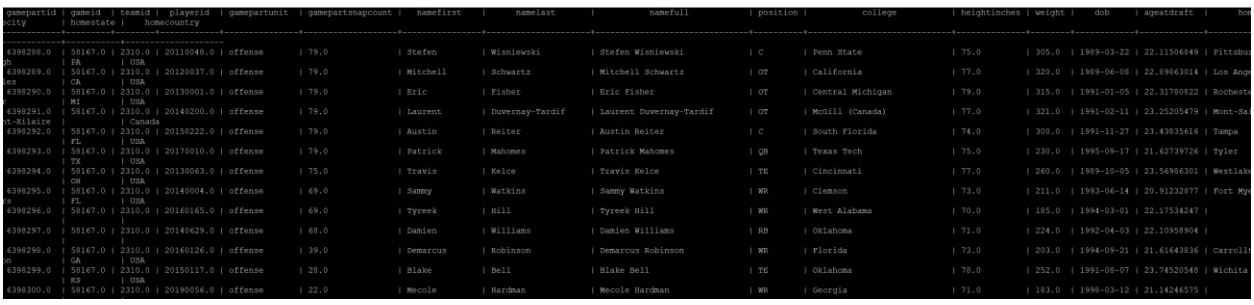
Copy data from CSV to table:

\COPY gameParticipation from gameParticipation.csv with csv header

```
su22adb20=>  \COPY gameParticipation FROM gameParticipation.csv with csv header
COPY 158211
```

Screenshot of the populated table:

SELECT * FROM gameParticipation;



Cardinality of the table:

SELECT COUNT(*) FROM tackles;

```
su22adb20=> SELECT COUNT(*) FROM gameParticipation
;
 count
--------
 158211
(1 row)
```

Primary Key:

ALTER TABLE gameParticipation add constraint pkey_gamePartId primary key(gamePartId);

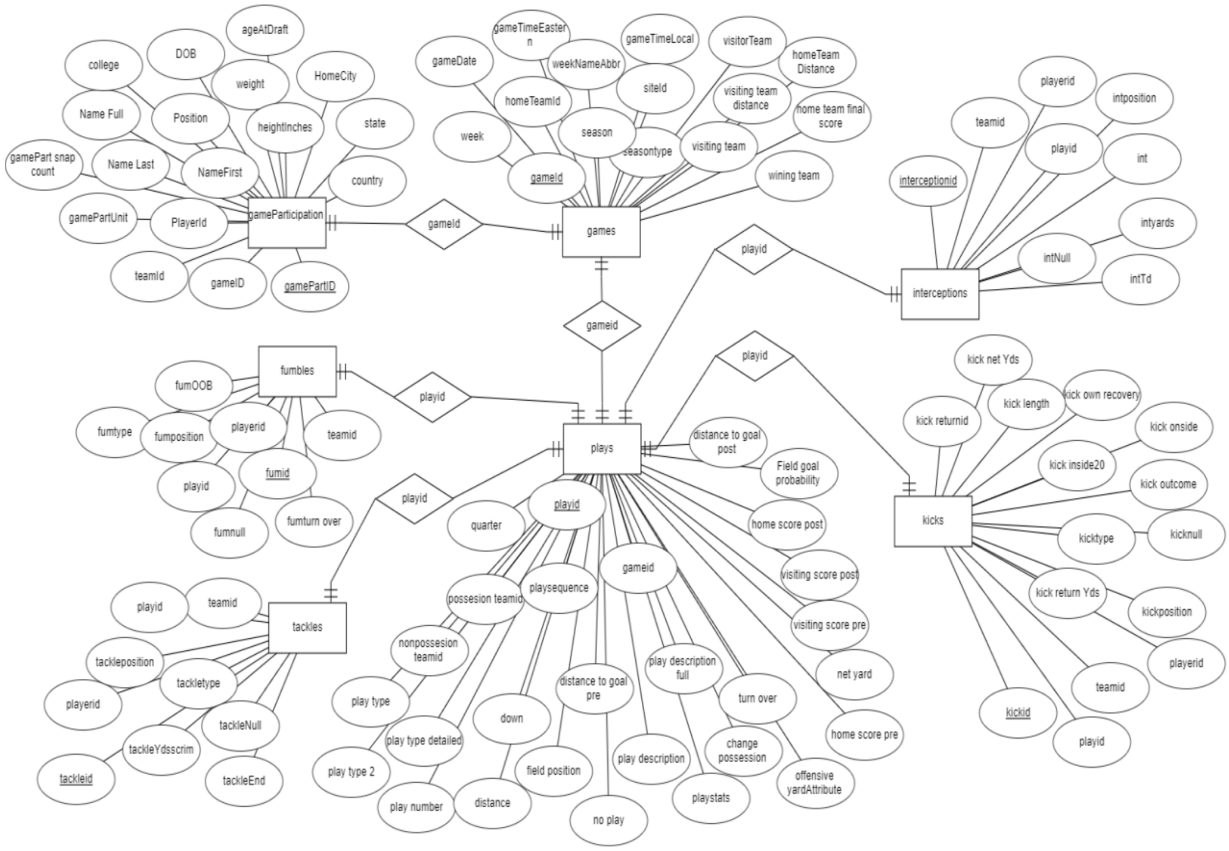
```
su22adb20=> ALTER TABLE gameParticipation add constraint pkey_gamePartId primary key(gamePartId);
ALTER TABLE
```

Foreign Key:

ALTER TABLE gameParticipation add constraint fk_gameParticipationid foreign key(gameid) references games(gameid);

```
su22adb20=> ALTER TABLE gameParticipation add constraint fk_gameParticipationid foreign key(gameid) references games(gameid);
ALTER TABLE
```

ER-Diagram:



Questions

Q1) Finding out the type and outcome of the kick with the longest length among all kicks.

Answer: The table that is used to find the answer is kicks.

The query type is Subquery with aggregate function which gives the below result:

```
SELECT kickType, kickOutcome FROM kicks WHERE kickLength = (SELECT MAX (kickLength) FROM kicks);
```

```
su22adb20=> SELECT kickType,kickOutcome FROM kicks WHERE kickLength = (SELECT MAX(kickLength) FROM kicks);
 kicktype | kickoutcome
----------+-------------
 kickoff  | touchback
(1 row)
```

Total number of rows returned = 1 row

Q2) Find the name of the players who play the type two point and belong to possession team ID 1540 with home score pre greater than 29.

Answer: The tables that are used to find the answer is gameParticipation, games and plays.

The query type is Inner Join query with more than 2 tables and LIMIT clause which gives the below result:

```
SELECT nameFull FROM gameParticipation gp
INNER JOIN games g ON g.gameId = gp.gameId
INNER JOIN plays p ON g.gameId = p.gameId
WHERE p.possesionteamid = 1540 and p.playtype = 'two-point' and p.homescorepre > 29;
```

```
     namefull
------------------
 Matt Flynn
 T.J. Lang
 Jarrett Boykin
 Bryan Bulaga
 David Bakhtiari
 Josh Sitton
 Corey Linsley
 Jordy Nelson
 Aaron Rodgers
 Randall Cobb
 Davante Adams
 Eddie Lacy
 Richard Rodgers
 Andrew Quarless
```

Total number of rows returned = 150 rows

Q3) Return the detailed play type for plays whose interception yards is greater than 100.

Answer: The tables that are used to find the answer is plays and interceptions.

The query type is Subquery with IN which gives the below result:

SELECT p.playtypedetailed FROM plays p WHERE p.playid IN (SELECT i.playid FROM interceptions i WHERE i.intyards>100);

```
su22adb20=>  select p.playtypedetailed from plays p where p.playid in (SELECT i.playid from interceptions i where i.intyards>100);
      playtypedetailed
------------------------------
 pass short left, intercepted
 pass, intercepted
 pass short right, intercepted
 pass short right, intercepted
 pass, intercepted
 pass short right, intercepted
 pass short right, intercepted
 pass deep left, intercepted
 pass, intercepted
 pass short right, intercepted
(10 rows)
```

Total number of rows returned = 10 rows

Q4) Provide the list of players' name and college attended for players whose date of birth is not populated

Answer: The table that is used to find the answer is gameParticipation

The query type is DISTINCT and NULL value which gives the below result:

SELECT DISTINCT(nameFull),college FROM gameParticipation WHERE dob IS NULL;

namefull	college
Lester Cotton	Alabama
Chris Johnson	North Alabama
Sean Smith	Dayton
Dravon Askew-Henry	West Virginia
Andrew Williams	Auburn
DeJuan Neal	Shepherd (WV)
Elijah Holyfield	Georgia
Connor Strachan	Boston
Cortrelle Simpson	Richmond
Kareem Orr	Chattanooga State
Marquise Copeland	Cincinnati
John Yarbrough	Richmond

Total number of rows returned = 172 rows

Q5) Find the count of Plays where forced fumbles happened along with tackles.

Answer: The table that is used to find the answer is fumbles, plays and tackles

The query type is GROUP BY, COUNT and HAVING which gives the below result:

```
SELECT f.fumtype,t.tackletype,count(*)
FROM fumbles f, tackles t, plays p
WHERE p.playid = f.playid AND
p.playid = t.playid
GROUP BY f.fumtype, t.tackletype HAVING f.fumtype = 'forced';
```

```
su22adb20=> SELECT f.fumtype,t.tackletype,count(*)
FROM fumbles f, tackles t, plays p
WHERE p.playid = f.playid AND
p.playid = t.playid
GROUP BY f.fumtype, t.tackletype HAVING f.fumtype = 'forced';
 fumtype | tackletype | count
---------+------------+-------
 forced  | assist     |   552
 forced  | for a loss |  2584
 forced  | solo       |  5037
(3 rows)
```

Total number of rows returned = 3 rows

Q6) Find the players' last names that begins with letter 'F' whose week name abbreviation is 'WC'.

Answer: The table that is used to find the answer is gameParticipation and games

The query type is JOIN and LIKE clause which gives the below result:

```
SELECT gp.nameLast FROM gameParticipation gp JOIN games g ON gp.gameid = g.gameid WHERE
g.weeknameabbr = 'WC' AND gp.nameLast LIKE 'F%';
```

```
  namelast
------------
 Folkerts
 Fanaika
 Floyd
 Fitzgerald
 Fells
 Frederick
 Fuller
 Fleener
 Foster
 Flacco
 Forsett
 Foote
 Fluellen
 Freeman
 Francois
 Flowers
 Folkerts
```

Total number of rows returned = 63 rows

Q7) List the first 50 plays and the detailed type of the plays

Answer: The table that is used to find the answer is plays

The query type is ORDER BY which gives the below result:

SELECT playid,playtypedetailed FROM plays ORDER BY playid,playtypedetailed ASC LIMIT 50;

```
su22adb20=> SELECT playid,playtypedetailed FROM plays ORDER BY playid,playtypedetailed ASC LIMIT 50;
 playid |  playtypedetailed
--------+---------------------
      1 | kickoff, returned
      2 | rush, left end
      3 | pass, complete
      4 | rush, left tackle
      5 | pass, incomplete
      6 | rush, up the middle
      7 | punt, returned
      8 | rush, right end
      9 | rush, left end
     10 | pass, incomplete
     11 | rush, right tackle
     12 | rush, right end
     13 | pass, incomplete
     14 | pass, incomplete
```

Total number of rows returned = 50 rows

Q8) Write an SQL view definition that displays each fumble type and, calculate and display the average of visiting score post for plays belonging to each fumble type.

Answer:

The table that is used to find the answer is plays and fumbles

The query type is View which gives the below result:

```
CREATE VIEW FumtypeAvgVisitingscorepostView AS
SELECT  f.fumtype, AVG(p.visitingscorepost) AS visiting_score_post
FROM plays p,fumbles f
where f.playid = p.playid
GROUP BY f.fumtype;
SELECT * FROM FumtypeAvgVisitingscorePostView;
```

```
CREATE VIEW
su22adb20=> SELECT * FROM FumtypeAvgVisitingscorePostView;
 fumtype   | visiting_score_post
-----------+---------------------
 forced    | 10.1837209302325581
 unforced  |  9.4960159362549801
(2 rows)
```

Total number of rows returned = 2 rows

Q9) List the top five colleges attended by the NFL players and their home city and home country.

Answer:

The table that is used to find the answer is plays and fumbles

The query type is GROUP BY and NOT NULL which gives the below result:

```
SELECT college,homecity,homecountry,count(college) FROM gameParticipation GROUP BY college, homecity, homecountry HAVING homecountry is NOT NULL ORDER BY COUNT DESC LIMIT 5;
```

```
     college      |   homecity   | homecountry | count
------------------+--------------+-------------+-------
 Miami (FL)       | Miami        | USA         |   704
 Texas            | Houston      | USA         |   497
 Louisiana State  | New Orleans  | USA         |   413
 USC              | Los Angeles  | USA         |   408
 Texas            | Dallas       | USA         |   372
(5 rows)
```

Total number of rows returned = 5 rows

Q10) List the full names, age at drafting and home state for the players who were drafted at age 24 and above and belong to the state of Florida.

Answer:

The table that is used to find the answer is gameParticipation

SELECT DISTINCT nameFull, ageAtDraft, homeState FROM gameParticipation WHERE ageAtDraft>24 AND homeState = 'FL';

namefull	ageatdraft	homestate
Brandon Dixon	24.04931507	FL
Brandon Doughty	24.57808219	FL
Brian Dixon	24.04931507	FL
Byron Pringle	24.4630137	FL
David Sims	24.5369863	FL
Donatello Brown	25.97260274	FL
Eddie Jackson	24.39726027	FL

Total number of rows returned = 27 rows