Milestone #3: Progress Report
NFL Play statistics Dataset
Submitted by - ***Chitradevi Maruthavanan***

Reasons to change the proposal from milestone #2:
Now, I am using mac OS since my old windows laptop was not working well. I installed the postgres in local machine and running the python script from terminal

Python Code for connection initialization and table creation:

```
import psycopg2
import pandas as pd
def initialize():
    connection = psycopg2.connect(
        user = "postgres", #username that you use
        password = "pwd", #password that you use
        host = "localhost",
        port = "5432",
        database = "postgres"
    )
    connection.autocommit = True
    return connection

def createTable(conn):
    with conn.cursor() as cursor:
        cursor.execute(f"""
            DROP TABLE IF EXISTS fumbles;
            CREATE TABLE fumbles (
                fumid           NUMERIC,
                playid          Numeric,
                teamid          NUMERIC,
                playerid        NUMERIC,
                fumposition     TEXT,
                fumtype         TEXT,
                fumoob          NUMERIC,
                fumturnover     FLOAT,
                 fumnull        NUMERIC
                 );
            ALTER TABLE fumbles ADD PRIMARY KEY (fumid);
        """)

    print(f"Created FUMBLES table")
def main():
    conn = initialize()
    createTable(conn)
```

The below screenshot is for the fumble table creation:

```
chitradevi@Chitradevis-MacBook-Pro intro_to_db % python3 db_project_3.py
Created FUMBLES table
```

The below Postgres SQL server output for the created fumbles table schema:

```
postgres-# \d fumbles
                    Table "public.fumbles"
    Column     |        Type        | Collation | Nullable | Default
---------------+--------------------+-----------+----------+---------
 fumid         | numeric            |           | not null |
 playid        | numeric            |           |          |
 teamid        | numeric            |           |          |
 playerid      | numeric            |           |          |
 fumposition   | text               |           |          |
 fumtype       | text               |           |          |
 fumoob        | numeric            |           |          |
 fumturnover   | double precision   |           |          |
 fumnull       | numeric            |           |          |
Indexes:
    "fumbles_pkey" PRIMARY KEY, btree (fumid)
```

Cardinality:

```
postgres=# select count(*) from fumbles;
 count
-------
     0
(1 row)
```

Copy command:

```
def insertTable(conn):
    pg_insert = '''COPY fumbles(fumid,playid,teamid,playerid,fumposition,fumtype,fumoob,fumturnover,fumnull)
        FROM '/tmp/DB Project files/fumbles.csv'
       DELIMITER ','
        CSV HEADER;'''
    with conn.cursor() as cursor:
        cursor.execute(pg insert)

        count = cursor.rowcount
        print(count, "Successfully inserted")
def main():
    conn = initialize()
    insertTable(conn)
```

The below screenshot is for the copy command

```
chitradevi@Chitradevis-MacBook-Pro intro_to_db % python3 db_project_3.py
14910 Successfully inserted
```

The below Postgres SQL server output for the inserted fumbles table :

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

Total number of rows in the fumbles table are:

```
postgres=# select count(*) from fumbles;
 count
-------
 14910
(1 row)
```

Questions

Q1) Finding out the DOB and highest visiting team final score of Steven Miller.

Answer:

```
def runQuery1(conn):
   select_Query = "select gp.nameFull,gp.dob,MAX(g.visitingteamfinalscore)AS highestVisitingteamFinalscore from games g INNER JOIN
gameParticipation gp ON g.gameid = gp.gameid  GROUP BY gp.nameFull,gp.dob Having gp.nameFull = 'Steven Miller'"
    highestVisitingteamFinalscore_df = pd.DataFrame(columns = ['nameFull','dob','highest visitingteamfinalscore'])

    with conn.cursor() as cursor:
        cursor.execute(select Query)
        records = cursor.fetchall()
        for row in records:
            output_df = {'nameFull': row[0],'dob': row[1], 'highestvisitingteamfinalscore ': row[2] }
            highestVisitingteamFinalscore df = pd.concat([highestVisitingteamFinalscore df,pd.DataFrame.from records([output df])])

        print(highestVisitingteamFinalscore df)

   def main():
    conn = initialize()
    runQuery1(conn)
```

Output:

```
[chitradevi@Chitradevis-MacBook-Pro ~ % python3 PSU_Coursework/intro_to_db/db_project_3.py
         nameFull         dob highestvisitingteamfinalscore
 0  Steven Miller  1991-03-23                            26
```

Q2) Finding out the lowest five tackle yds scrim with tackle

Answer:

```
def runQuery2(conn):
select_Query = "select distinct  tackletype,tackleydsscrim from tackles order by tackleydsscrim,tackletype desc limit 5"
tackletype_df = pd.DataFrame(columns = ['tackletype','tackleydscrim'])

with conn.cursor() as cursor:
cursor.execute(select Query)
records = cursor.fetchall()
for row in records:
output df = {'tackletype': row[0],'tackleydscrim': row[1]}
tackletype_df = pd.concat([tackletype_df,pd.DataFrame.from_records([output_df])])

print(tackletype_df)
```

Output:

```
[chitradevi@Chitradevis-MacBook-Pro intro_to_db % python3 db_project_3.py
    tackletype tackleydscrim
0         solo          -1.0
0  for a loss          -1.0
0      assist          -1.0
0         solo         -10.0
0  for a loss         -10.0
```

Q3) Find game participant name, unit and snapcount for player who lives in Vermont

Answer:

```
def runQuery3(conn):
select Query = "select gameid,nameFirst,gamePartUnit,gamepartSnapCount from gameparticipation where homeState ='VT'"
vermontgameParticipant_df = pd.DataFrame(columns = ['gameid','nameFirst','gamePartUnit','gamepartSnapCount'])

with conn.cursor() as cursor:
cursor.execute(select_Query)
records = cursor.fetchall()
for row in records:
output_df = {'gameid': row[0],'nameFirst': row[1],'gamePartUnit' : row[2],'gamepartSnapCount': row[3]}
vermontgameParticipant df = pd.concat([vermontgameParticipant df,pd.DataFrame.from records([output df])])

print(vermontgameParticipant df)
```

Output:

```
chitradevi@Chitradevis-MacBook-Pro intro_to_db % python3
  gameid nameFirst   gamePartUnit gamepartSnapCount
0  56466     Jason        offense                21
0  56435     Jason        offense                31
0  56453     Jason        offense                18
0  56466     Jason  special teams                 1
0  56435     Jason  special teams                 2
0  56453     Jason  special teams                 1
chitradevi@Chitradevis-MacBook-Pro intro_to_db %
```