**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:**

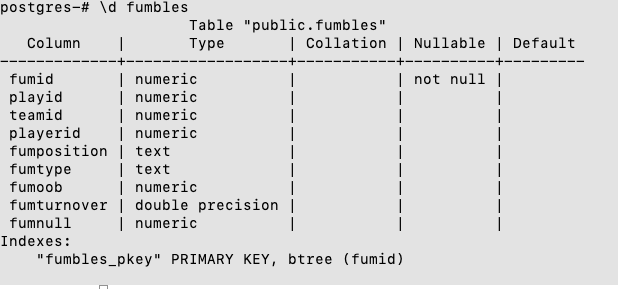
import psycopg2  
import pandas as pd  
def initialize():  
 connection = psycopg2.connect(  
 user = "postgres", *#username that you use* password = "pwd", *#password that you use* host = "localhost",   
 port = "5432",   
 database = "postgres"  
 )  
 connection.autocommit = True  
 return connection  
 def createTable(conn):  
 with conn.cursor() as cursor:  
 cursor.execute(f"""  
 DROP TABLE IF EXISTS fumbles;  
 CREATE TABLE fumbles (  
 fumid NUMERIC,  
 playid Numeric,S  
 teamid NUMERIC,  
 playerid NUMERIC,   
 fumposition TEXT,   
 fumtype TEXT,   
 fumoob NUMERIC,   
 fumturnover FLOAT,SSS  
 fumnull NUMERIC  
 );  
 ALTER TABLE fumbles ADD PRIMARY KEY (fumid);  
 """)  
  
 print(f"Created FUMBLES table")

def main():  
 conn = initialize()  
 createTable(conn)

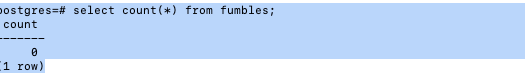
The below screenshot is for the fumble table creation:

****

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



Cardinality:

****

**Copy command:**

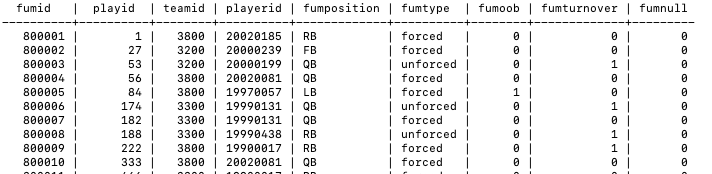
def insertTable(conn):  
 pg\_insert = '''COPY fumbles(fumid,playid,teamid,playerid,fumposition,fumtype,fumoob,fumturnover,fumnull)   
 FROM '/tmp/DB\_Project\_files/fumbles.csv'  
 DELIMITER ','  
 CSV HEADER;'''  
 with conn.cursor() as cursor:  
 cursor.execute(pg\_insert)  
  
 count = cursor.rowcount  
 print(count, "Successfully inserted")

def main():  
 conn = initialize()  
 insertTable(conn)

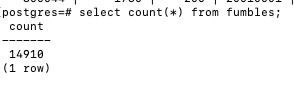
**The below screenshot is for the copy command**

****

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

****

Total number of rows in the fumbles table are:

****

**Questions**

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

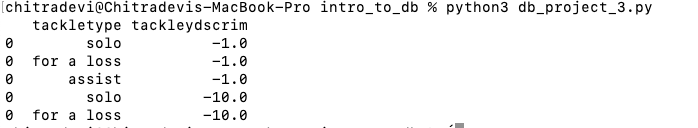
**Answer:**

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:**

****

**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)

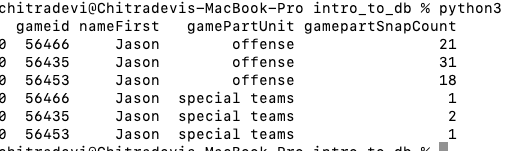
**Output:**

**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:**

****