CompeteNow: a Tournament Organization Platform

Milestone: Python Application
Group 24

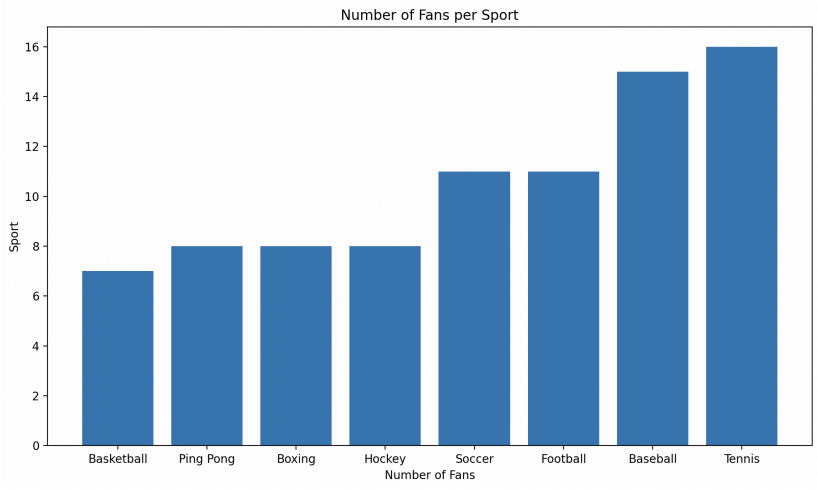
Charles Schatmeyer
716-713-1492
schatmeyer.c@northeastern.edu

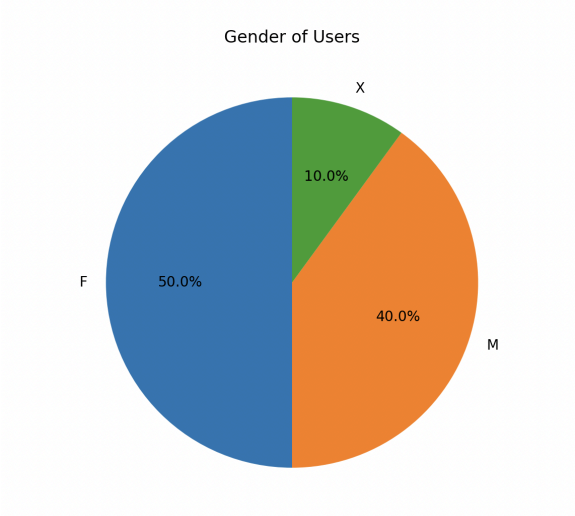
Percentage of Effort Contributed by Student1: 100%

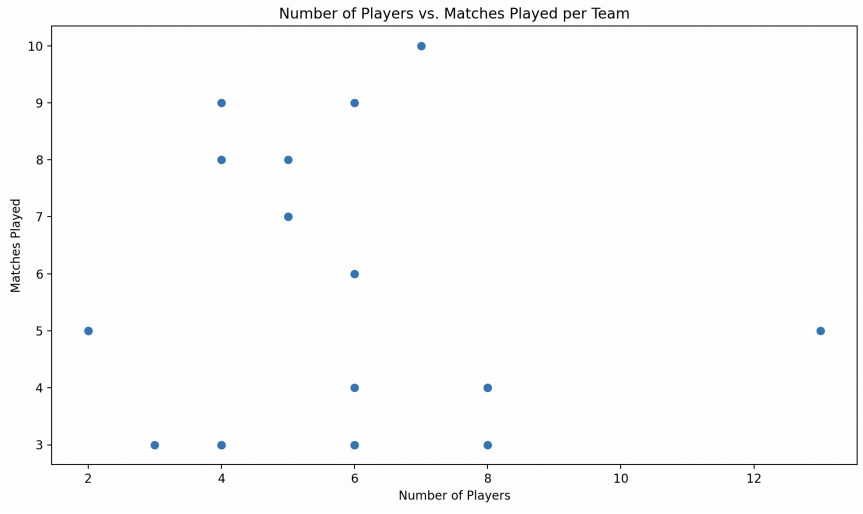
Signature of Student 1: 

Submission Date: 11/22/2024

Charts:


Number of Fans per Sport
Sport
Number of Fans
Basketball
Ping Pong
Boxing
Hockey
Soccer
Football
Baseball
Tennis
0
2
4
6
8
10
12
14
16


Gender of Users
X
10.0%
F
50.0%
40.0%
M


Number of Players vs. Matches Played per Team
Matches Played
Number of Players
3
4
5
6
7
8
9
10
2
4
6
8
10
12

Code:

```
# IE6700 Python App
# Competition Application
# Charlie Schatmeyer

import mysql.connector
import matplotlib.pyplot as plt


# Print Query Results
def print_q():
   rows = cursor.fetchall()
   for row in rows:
       print(row)


# Simple Query
def simple_q():
   cursor.execute("SELECT * FROM individuals_match")


def agg_q():
   cursor.execute("select gender, COUNT(*) as count "
                  "from user "
                  "group by gender")

   rows = cursor.fetchall()
   labels = [row[0] for row in rows]
   sizes = [row[1] for row in rows]

   plt.figure(figsize=(8, 6))
   plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90)
   plt.title("Gender of Users")
   plt.show()


def join_q():
   cursor.execute("select name, COUNT(*) as number_of_fans "
                  "from sport s "
                  "right outer join spectator_follows sf "
                  "on s.sportID = sf.sport "
                  "group by name "
                  "order by number_of_fans")

   rows = cursor.fetchall()
   names = [row[0] for row in rows]
   fans = [row[1] for row in rows]
```

```
    plt.figure(figsize=(10, 6))
    plt.bar(names, fans)
    plt.xlabel("Number of Fans")
    plt.ylabel("Sport")
    plt.title("Number of Fans per Sport")
    plt.tight_layout()
    plt.show()


def nest_q():
    cursor.execute("select name, hours "
                   "from venue "
                   "where venueID in ("
                       "select venueID "
                       "from match_gen "
                       "where tournamentID in ("
                           "select tournamentID "
                           "from tournament "
                           "where name = 'Championship 1'"
                       ")"
                   ")")


def corr_q():
    cursor.execute("select matchID, time "
                   "from match_gen "
                   "where scoreAlpha + scoreAlpha > ("
                       "select avg(scoreAlpha + scoreBeta) "
                       "from match_gen"
                   ")")


def all_q():
    cursor.execute("select name, age "
                   "from user "
                   "where age >= ALL ("
                       "select age "
                       "from user"
                   ")")


def union_q():
    cursor.execute("select u.name as 'Scored Zero Points' "
                   "from user u "
                   "join player p on u.userID = p.playerID "
                   "where p.playerID in ("
                       "select playerAlpha "
                       "from individuals_match "
                       "where matchID in ("
```

```
                            "select matchID "
                            "from match_gen "
                            "where scoreAlpha = 0"
                        ")"
                    ")"
                    " union "
                    "select u.name as 'Scored Zero Points' "
                    "from user u "
                    "join player p on u.userID = p.playerID "
                    "where p.playerID in ("
                        "select playerBeta "
                        "from individuals_match "
                        "where matchID in ("
                            "select matchID "
                            "from match_gen "
                            "where scoreBeta = 0 "
                        ")"
                    ")")


def sub_q():
   cursor.execute("select t.name,"
                           "(select count(tp.playerID) from team_player tp where
tp.teamID = t.teamID group by t.teamID) as 'Num Players',"
                           "(select count(*) from teams_match tm where
tm.teamBeta = t.teamID or tm.teamAlpha = t.teamID) as 'Matches Played' "
                    "from team t")

   rows = cursor.fetchall()
   num_players = [row[1] for row in rows]  # number of players per team
   matches_played = [row[2] for row in rows]  # matches played per team

   # Generate the scatter plot
   plt.figure(figsize=(10, 6))
   plt.scatter(num_players, matches_played)
   plt.xlabel("Number of Players")
   plt.ylabel("Matches Played")
   plt.title("Number of Players vs. Matches Played per Team")
   plt.tight_layout()
   plt.show()


# Connect to mysql
connection = mysql.connector.connect(
   host="localhost",
   port=3307,
   user="root",
   password="password",
   database="competition"
```

```
)
cursor = connection.cursor()

# Setup UI While Loop
continue_var = True

# UI While Loop
while continue_var:
   qn = input("Which query?"
              "\n1 - Simple"
              "\n2 - Aggregate + Pie Chart (Gender of Users)"
              "\n3 - Inner/Outer Join + Histogram (# of Fans of Sport)"
              "\n4 - Nested"
              "\n5 - Correlated"
              "\n6 - >=ALL/>ANY/Exists/Not Exists"
              "\n7 - Union"
              "\n8 - Sub-queries (Select, From) + Scatter Plot (Team Players vs
Matches)"
              "\n9 - Quit"
              "\n")

   if qn == '1':
       simple_q()
   elif qn == '2':
       agg_q()
   elif qn == '3':
       join_q()
   elif qn == '4':
       nest_q()
   elif qn == '5':
       corr_q()
   elif qn == '6':
       all_q()
   elif qn == '7':
       union_q()
   elif qn == '8':
       sub_q()
   else:
       continue_var = False
   print_q()

# Close the cursor and connection
cursor.close()
connection.close()
```