Sam Logsdon - Assignment 3

October 16, 2019

[ ]:
```
import sqlite3
conn = sqlite3.connect("database.sqlite")

def run_query(sql: str):
    return conn.execute(sql).fetchall()
```

0.0.1 Problem 1

[ ]:
```
sql = '''
SELECT player_name 'Player Name', STRFTIME("%Y-%m-%d", birthday) 'Birthday'
FROM Player
WHERE STRFTIME('%Y', birthday) BETWEEN '1987' AND '1990'
ORDER BY Birthday;'''

rows = run_query(sql)
print(f'{"Player Name":20} | Birthday')
for row in rows:
    print(f'{row[0]:20.20} | {row[1]}')
```

0.0.2 Problem 2

[ ]:
```
sql = '''
SELECT C.name 'Country', L.name 'League Name', SUM(M.home_team_goal + M.
 ↪away_team_goal) "Total Goals Scored"
FROM Country C
         JOIN League L on C.id = L.country_id
         JOIN Match M on C.id = M.country_id
GROUP BY L.name, C.name
ORDER BY 3 DESC;'''
rows = run_query(sql)
print(f'{"Country":15}   {"League Name":30}   {"Total Goals Scored"}')
for row in rows:
    print(f'{row[0]:15} | {row[1]:30} | {row[2]:4}')
```

0.0.3 Problem 3

```
[ ]: sql = '''
SELECT team_long_name "Team Long Name", ROUND(AVG(((buildUpPlaySpeed +
                        buildUpPlayDribbling +
                        buildUpPlayPassing +
                        chanceCreationPassing +
                        chanceCreationCrossing +
                        chanceCreationShooting +
                        defenceAggression +
                        defencePressure +
                        defenceTeamWidth )/ 9.0)), 2) "Team Stats"
FROM Team
         JOIN Team_Attributes TA on Team.team_api_id = TA.team_api_id
GROUP BY 1
HAVING "Team Stats" NOT NULL
ORDER BY 2 desc;'''
rows = run_query(sql)
print(f'{"Team Long Name":35}  {"Team Stats"}')
for row in rows:
    print(f'{row[0]:35} | {row[1]:<5}')
```

0.0.4 Problem 4

```
[ ]: sql = '''
WITH PA as (
    SELECT max(date), *
    FROM Player_Attributes
    GROUP BY player_api_id
),
     combined_matches(player_name, player_id, team_name, team_id) as (
         SELECT DISTINCT Player.player_name,
                Player.player_api_id,
                T.team_long_name,
                T.team_api_id

         from Player
                  JOIN Match M ON
                 Player.player_api_id
                 IN (
                     VALUES (home_player_1),
                            (home_player_2),
                            (home_player_3),
                            (home_player_4),
                            (home_player_5),
                            (home_player_6),
                            (home_player_6),
```

```
                            (home_player_7),
                            (home_player_8),
                            (home_player_9),
                            (home_player_10),
                            (home_player_11)
                    )
                     JOIN Team T
                          ON (M.home_team_api_id = T.team_api_id)
            GROUP BY team_api_id, player_api_id
            UNION
            SELECT DISTINCT Player.player_name,
                   Player.player_api_id,
                   T.team_long_name,
                   T.team_api_id

            from Player
                     JOIN Match M ON
                    Player.player_api_id
                    IN (
                        VALUES (away_player_1),
                               (away_player_2),
                               (away_player_3),
                               (away_player_4),
                               (away_player_5),
                               (away_player_6),
                               (away_player_6),
                               (away_player_7),
                               (away_player_8),
                               (away_player_9),
                               (away_player_10),
                               (away_player_11)
                    )
                     JOIN Team T
                          ON M.away_team_api_id = T.team_api_id
            GROUP BY team_api_id, player_api_id
        )
SELECT team_name 'Team Name',
            count(*) 'Number of Players',
           round(avg(overall_rating), 2) 'Player Attribute Average'

FROM combined_matches
             JOIN PA ON player_id = player_api_id
GROUP BY team_id
ORDER BY
             "Player Attribute Average" DESC
LIMIT 5;
'''
```

```
rows = run_query(sql)
print(f'{"Team Name":30}   {"Number of Players":20}   {"Player Attribute␣
 ↪Average"}')
for row in rows:
    print(f'{row[0]:30} | {row[1]:<20} | {row[2]:<5}')
```

0.0.5 Problem 5

[ ]:
```
sql = '''
SELECT STRFTIME('%d/%m/%Y', date) 'Date (dd/mm/yy)',
       season Season,
       L.name 'League Name',
       MAX(M.home_team_goal + M.away_team_goal) 'Goals Scored'
FROM Match M
         JOIN League L on m.league_id = L.id
GROUP BY 2, 3
ORDER BY 3, 2;
'''

rows = run_query(sql)
print(f'{"Date (dd/mm/yy)":15}   {"Season":10}   {"League Name":25}   {"Goals␣
 ↪Scored"}')
for row in rows:
    print(f'{row[0]:15} | {row[1]:10} | {row[2]:25} | {row[3]:<2}')
```

0.0.6 Graduate Student Task

[ ]:
```
sql = '''
WITH home_scores AS
         (
             SELECT season, SUM(home_team_goal) goals, home_team_api_id,␣
 ↪league_id
             FROM Match
             WHERE season LIKE '%2008/2009%'
             GROUP BY home_team_api_id),
     away_scores AS
         (SELECT season, SUM(away_team_goal) goals, away_team_api_id, league_id
          FROM Match
          WHERE season LIKE '%2008/2009%'
          GROUP BY away_team_api_id),
     scores AS (
         SELECT (home_scores.goals + away_scores.goals) goals,
                home_team_api_id team_id,
```

```
                away_scores.league_id league_id,
                away_scores.season season
        FROM home_scores
                 JOIN away_scores ON home_scores.home_team_api_id = away_scores.
 ↪away_team_api_id
    ),
    ranked_scores as (
        SELECT scores.goals,
               scores.league_id,
               scores.team_id,
               scores.season,
               rank() OVER
                   (PARTITION BY league_id ORDER BY goals Desc) as rank
        FROM scores)
SELECT season Season, L.name League, ranked_scores.rank Rank, T.team_long_name␣
 ↪'Team Name', ranked_scores.goals 'Goals Scored'
from ranked_scores
JOIN League L ON ranked_scores.league_id = L.id
JOIN Team T ON ranked_scores.team_id = T.team_api_id
WHERE rank <= 5;
'''
rows = run_query(sql)
print(f'{"Season":10}   {"League":25}   {"Rank":5}   {"Team Name":25}   {"Goals␣
 ↪Scored"}')
for row in rows:
    print(f'{row[0]:10} | {row[1]:25} | {row[2]:<5} | {row[3]:25} | {row[4]:<3}')
```