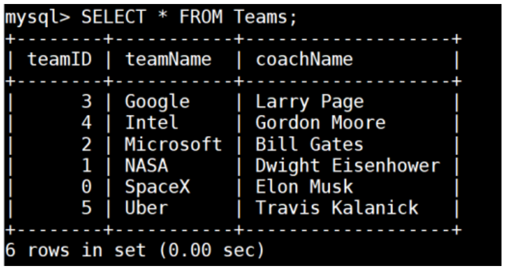
Andrea Gray
Files and Database Systems CS317
Assignment #3 – **Soccer Database**

[1] *Create a database, like the Access database, with the following criteria:*

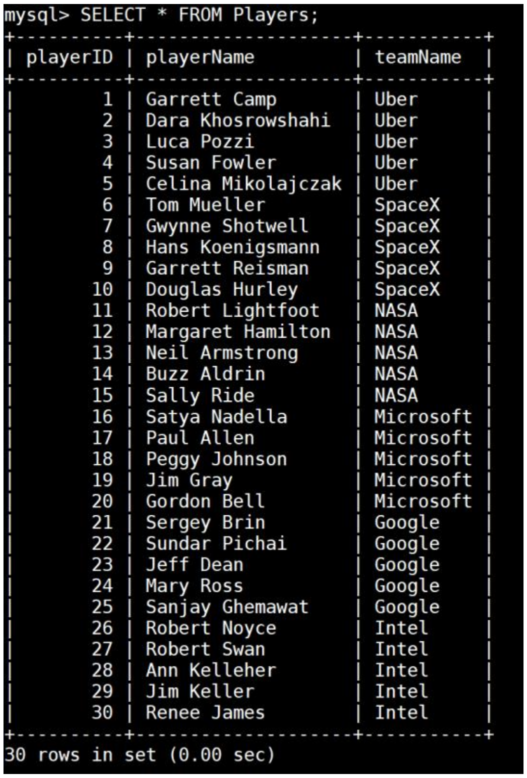
a. *There must be at least six teams*



```
mysql> SELECT * FROM Teams;
+--------+-----------+-------------------+
| teamID | teamName  | coachName         |
+--------+-----------+-------------------+
|      3 | Google    | Larry Page        |
|      4 | Intel     | Gordon Moore      |
|      2 | Microsoft | Bill Gates        |
|      1 | NASA      | Dwight Eisenhower |
|      0 | SpaceX    | Elon Musk         |
|      5 | Uber      | Travis Kalanick   |
+--------+-----------+-------------------+
6 rows in set (0.00 sec)
```

b. *Each team must have at least five players*

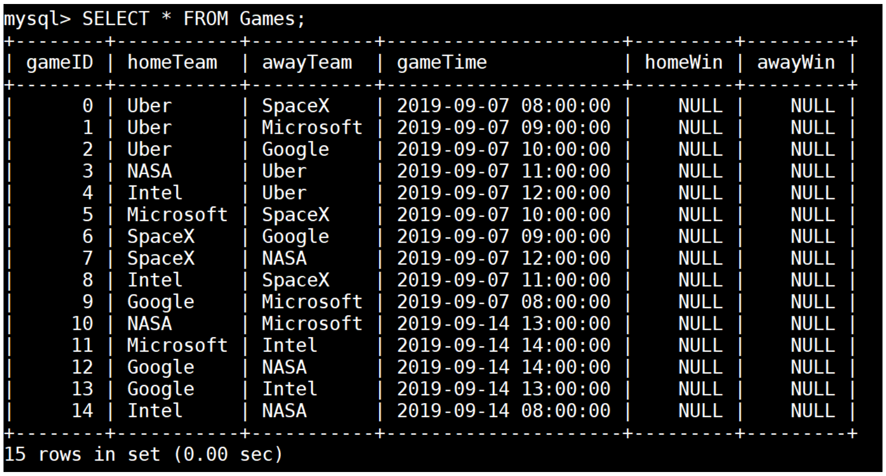
The player name is one key and is not separated into two separate keys for first and last name. This design feature is supported by the assignment questions not requesting a specific query or table creation that uses specifically only the first or last name of a player.



```
mysql> SELECT * FROM Players;
+----------+--------------------+-----------+
| playerID | playerName         | teamName  |
+----------+--------------------+-----------+
|        1 | Garrett Camp       | Uber      |
|        2 | Dara Khosrowshahi  | Uber      |
|        3 | Luca Pozzi         | Uber      |
|        4 | Susan Fowler       | Uber      |
|        5 | Celina Mikolajczak | Uber      |
|        6 | Tom Mueller        | SpaceX    |
|        7 | Gwynne Shotwell    | SpaceX    |
|        8 | Hans Koenigsmann   | SpaceX    |
|        9 | Garrett Reisman    | SpaceX    |
|       10 | Douglas Hurley     | SpaceX    |
|       11 | Robert Lightfoot   | NASA      |
|       12 | Margaret Hamilton  | NASA      |
|       13 | Neil Armstrong     | NASA      |
|       14 | Buzz Aldrin        | NASA      |
|       15 | Sally Ride         | NASA      |
|       16 | Satya Nadella      | Microsoft |
|       17 | Paul Allen         | Microsoft |
|       18 | Peggy Johnson      | Microsoft |
|       19 | Jim Gray           | Microsoft |
|       20 | Gordon Bell        | Microsoft |
|       21 | Sergey Brin        | Google    |
|       22 | Sundar Pichai      | Google    |
|       23 | Jeff Dean          | Google    |
|       24 | Mary Ross          | Google    |
|       25 | Sanjay Ghemawat    | Google    |
|       26 | Robert Noyce       | Intel     |
|       27 | Robert Swan        | Intel     |
|       28 | Ann Kelleher       | Intel     |
|       29 | Jim Keller         | Intel     |
|       30 | Renee James        | Intel     |
+----------+--------------------+-----------+
30 rows in set (0.00 sec)
```

c. *There must be games and a game schedule such that each team plays each other team exactly once*

The games to be played along with their scheduled date and time are all elements of the Games table. This table also contains an area for logging who won the game once it has been played (which will be in #5).



```
mysql> SELECT * FROM Games;
+--------+-----------+-----------+---------------------+---------+---------+
| gameID | homeTeam  | awayTeam  | gameTime            | homeWin | awayWin |
+--------+-----------+-----------+---------------------+---------+---------+
|      0 | Uber      | SpaceX    | 2019-09-07 08:00:00 |    NULL |    NULL |
|      1 | Uber      | Microsoft | 2019-09-07 09:00:00 |    NULL |    NULL |
|      2 | Uber      | Google    | 2019-09-07 10:00:00 |    NULL |    NULL |
|      3 | NASA      | Uber      | 2019-09-07 11:00:00 |    NULL |    NULL |
|      4 | Intel     | Uber      | 2019-09-07 12:00:00 |    NULL |    NULL |
|      5 | Microsoft | SpaceX    | 2019-09-07 10:00:00 |    NULL |    NULL |
|      6 | SpaceX    | Google    | 2019-09-07 09:00:00 |    NULL |    NULL |
|      7 | SpaceX    | NASA      | 2019-09-07 12:00:00 |    NULL |    NULL |
|      8 | Intel     | SpaceX    | 2019-09-07 11:00:00 |    NULL |    NULL |
|      9 | Google    | Microsoft | 2019-09-07 08:00:00 |    NULL |    NULL |
|     10 | NASA      | Microsoft | 2019-09-14 13:00:00 |    NULL |    NULL |
|     11 | Microsoft | Intel     | 2019-09-14 14:00:00 |    NULL |    NULL |
|     12 | Google    | NASA      | 2019-09-14 14:00:00 |    NULL |    NULL |
|     13 | Google    | Intel     | 2019-09-14 13:00:00 |    NULL |    NULL |
|     14 | Intel     | NASA      | 2019-09-14 08:00:00 |    NULL |    NULL |
+--------+-----------+-----------+---------------------+---------+---------+
15 rows in set (0.00 sec)
```

d. *The schedule must associate two teams with each game and designate one of the two as the host and therefore responsible for bringing the after-game snacks*

The host is identified in the query above under the column entitled: `homeTeam`

e. *Assignment as the host must be evenly distributed to the teams such that the difference between host games of team with most games as host and host games of team with fewest games as host is not bigger than two, but ideally being one.*

	Times as the Host	Times as the Visitor
Uber	3	2
NASA	2	3
Intel	3	2
Microsoft	2	3
SpaceX	2	3
Google	3	2

As can be seen from the table above, the deviation between the most and fewest times a team plays as the host team is no greater than one.

f. *Tables must not replicate data unnecessarily, which is to say that any data that appears in more than one table that must be a primary or candidate key in at least one table and represent a foreign key in any other table where it appears.*

The only data that is replicated between the tables is the name of the team which is the primary key in the Teams table (as `teamName)` and a foreign key in the other two tables: Games (as `homeTeam` and `awayTeam`) and Players (as `teamName`).

[2] *Submit complete sql code in the form of a file with a .sql suffix, that will create and populate all of your soccer league tables*

The .sql file is attached along with this document under the file name: `Gray_HW3_Q2_CS317.sql`

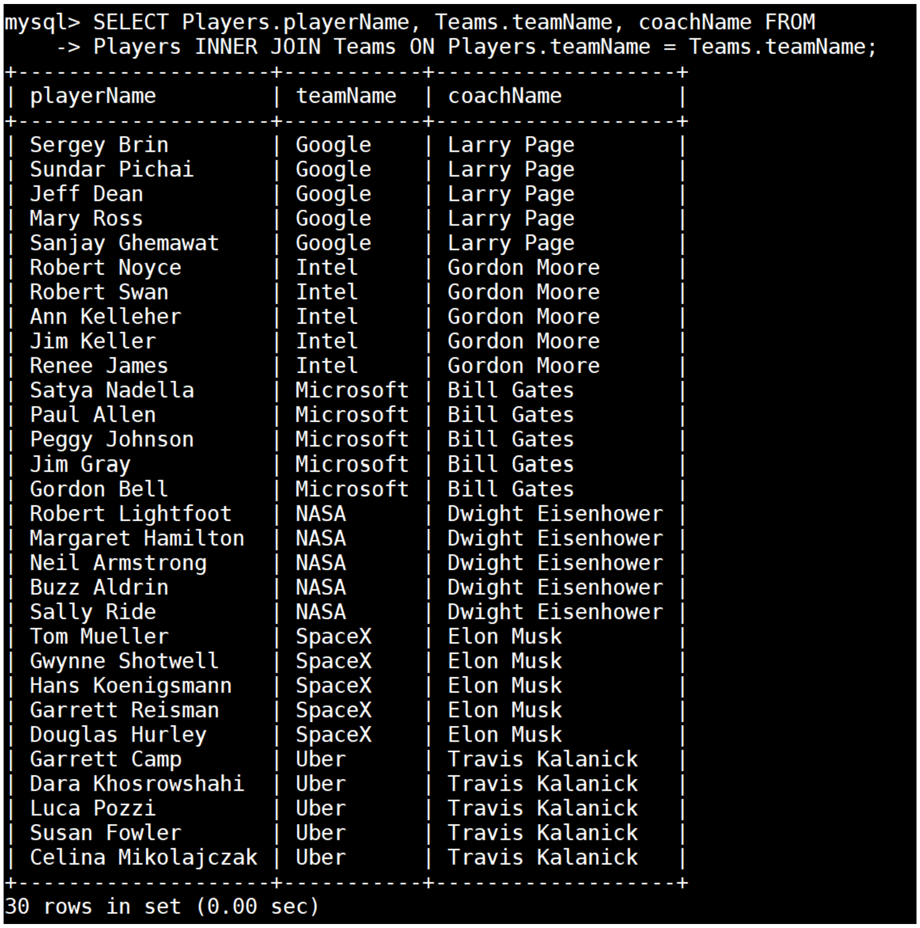
[3] *Submit complete sql code in the form of a file with a .sql suffix that will remove all of your soccer league tables*

The .sql file is attached along with this document under the file name: `Gray_HW2_Q3_CS317.sql`

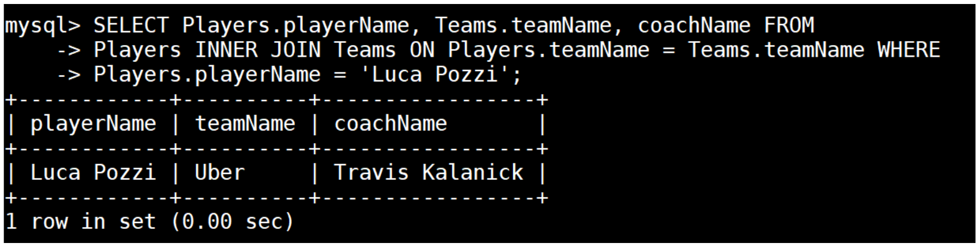
[4] *Put the following queries in a third .sql file and show output*

a. *A query generating a table with player name, team name, and team coach name*

Since the question is vague on whether it wants a table for all players or a specific player, both are included in the third .sql file (named: `Gray_HW3_Q4_CS317.sql`) and within the outputs below. The first output is for all players and the second is for a specific player, Luca Pozzi.



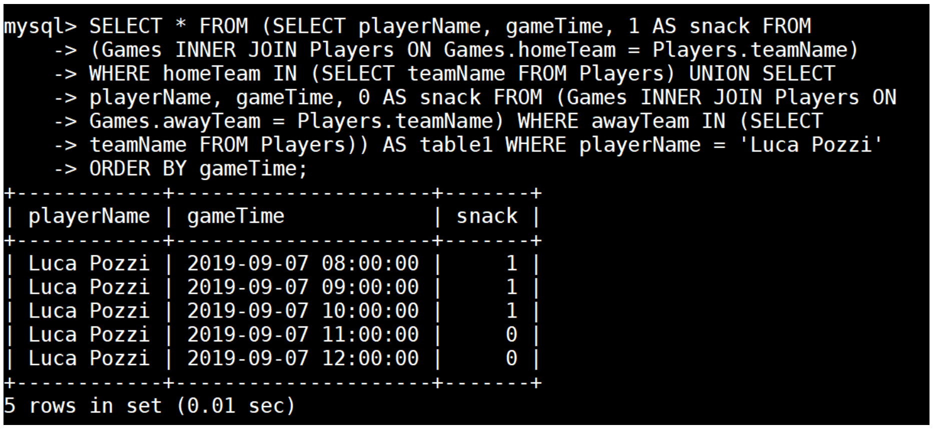
```
mysql> SELECT Players.playerName, Teams.teamName, coachName FROM
    -> Players INNER JOIN Teams ON Players.teamName = Teams.teamName;
+--------------------+-----------+-------------------+
| playerName         | teamName  | coachName         |
+--------------------+-----------+-------------------+
| Sergey Brin        | Google    | Larry Page        |
| Sundar Pichai      | Google    | Larry Page        |
| Jeff Dean          | Google    | Larry Page        |
| Mary Ross          | Google    | Larry Page        |
| Sanjay Ghemawat    | Google    | Larry Page        |
| Robert Noyce       | Intel     | Gordon Moore      |
| Robert Swan        | Intel     | Gordon Moore      |
| Ann Kelleher       | Intel     | Gordon Moore      |
| Jim Keller         | Intel     | Gordon Moore      |
| Renee James        | Intel     | Gordon Moore      |
| Satya Nadella      | Microsoft | Bill Gates        |
| Paul Allen         | Microsoft | Bill Gates        |
| Peggy Johnson      | Microsoft | Bill Gates        |
| Jim Gray           | Microsoft | Bill Gates        |
| Gordon Bell        | Microsoft | Bill Gates        |
| Robert Lightfoot   | NASA      | Dwight Eisenhower |
| Margaret Hamilton  | NASA      | Dwight Eisenhower |
| Neil Armstrong     | NASA      | Dwight Eisenhower |
| Buzz Aldrin        | NASA      | Dwight Eisenhower |
| Sally Ride         | NASA      | Dwight Eisenhower |
| Tom Mueller        | SpaceX    | Elon Musk         |
| Gwynne Shotwell    | SpaceX    | Elon Musk         |
| Hans Koenigsmann   | SpaceX    | Elon Musk         |
| Garrett Reisman    | SpaceX    | Elon Musk         |
| Douglas Hurley     | SpaceX    | Elon Musk         |
| Garrett Camp       | Uber      | Travis Kalanick   |
| Dara Khosrowshahi  | Uber      | Travis Kalanick   |
| Luca Pozzi         | Uber      | Travis Kalanick   |
| Susan Fowler       | Uber      | Travis Kalanick   |
| Celina Mikolajczak | Uber      | Travis Kalanick   |
+--------------------+-----------+-------------------+
30 rows in set (0.00 sec)
```



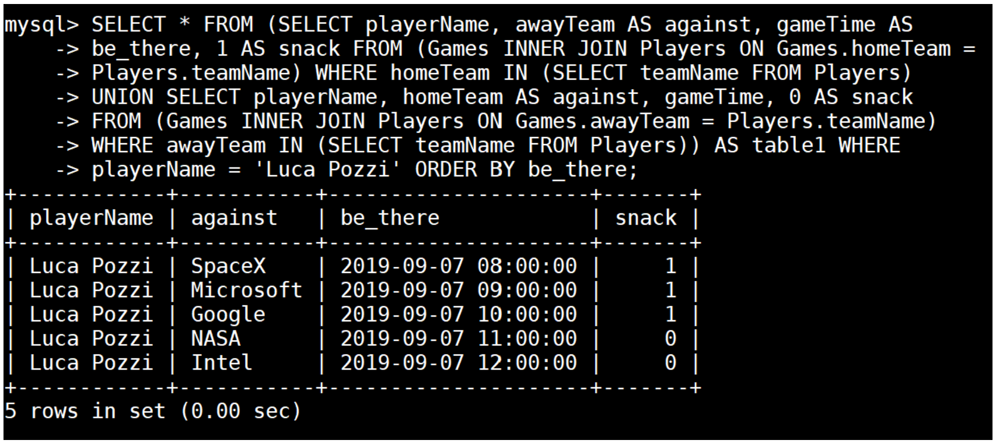
```
mysql> SELECT Players.playerName, Teams.teamName, coachName FROM
    -> Players INNER JOIN Teams ON Players.teamName = Teams.teamName WHERE
    -> Players.playerName = 'Luca Pozzi';
+------------+----------+-----------------+
| playerName | teamName | coachName       |
+------------+----------+-----------------+
| Luca Pozzi | Uber     | Travis Kalanick |
+------------+----------+-----------------+
1 row in set (0.00 sec)
```

b. *Queries capable of generating the tables shown on the assignment expectations document*

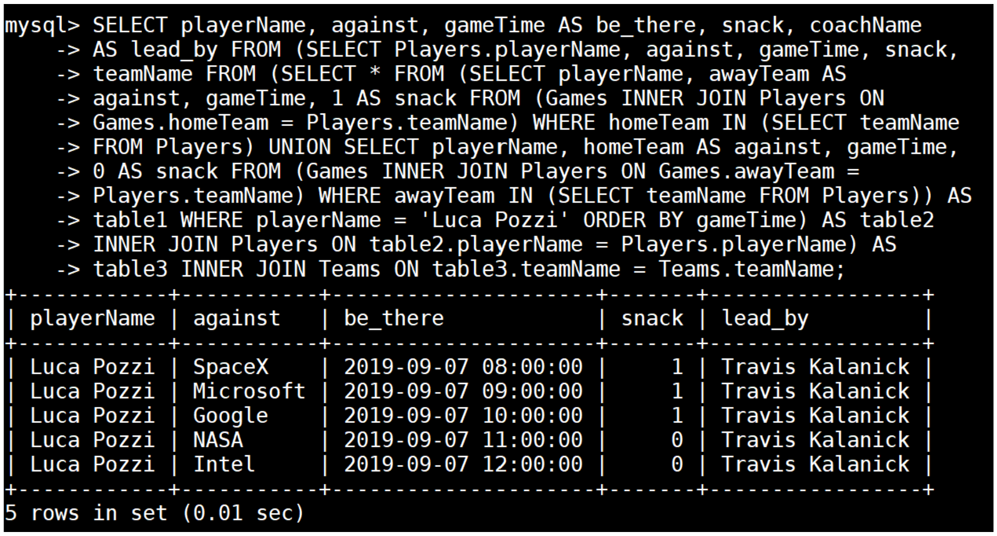
The queries for the regeneration of the tables shown are shown in the attached file (named: `Gray_HW3_Q4_CS317.sql`) with their respective outputs shown below.



```
mysql> SELECT * FROM (SELECT playerName, gameTime, 1 AS snack FROM
    -> (Games INNER JOIN Players ON Games.homeTeam = Players.teamName)
    -> WHERE homeTeam IN (SELECT teamName FROM Players) UNION SELECT
    -> playerName, gameTime, 0 AS snack FROM (Games INNER JOIN Players ON
    -> Games.awayTeam = Players.teamName) WHERE awayTeam IN (SELECT
    -> teamName FROM Players)) AS table1 WHERE playerName = 'Luca Pozzi'
    -> ORDER BY gameTime;
+------------+---------------------+-------+
| playerName | gameTime            | snack |
+------------+---------------------+-------+
| Luca Pozzi | 2019-09-07 08:00:00 |     1 |
| Luca Pozzi | 2019-09-07 09:00:00 |     1 |
| Luca Pozzi | 2019-09-07 10:00:00 |     1 |
| Luca Pozzi | 2019-09-07 11:00:00 |     0 |
| Luca Pozzi | 2019-09-07 12:00:00 |     0 |
+------------+---------------------+-------+
5 rows in set (0.01 sec)
```



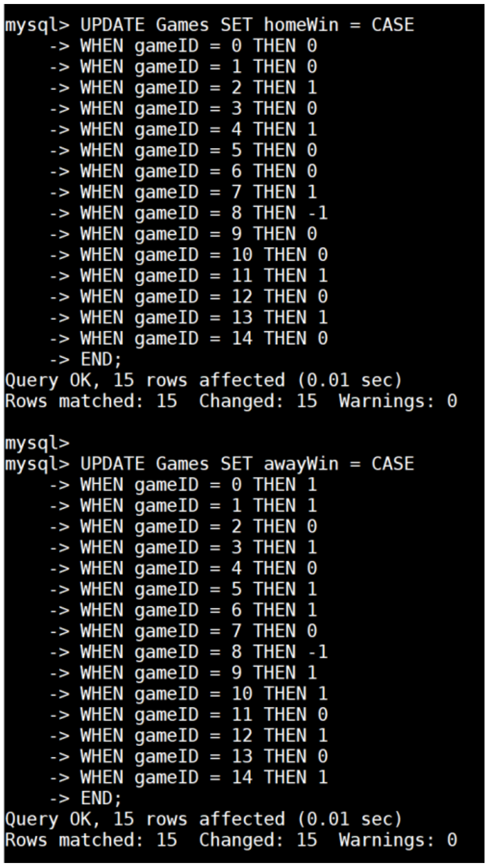
```
mysql> SELECT * FROM (SELECT playerName, awayTeam AS against, gameTime AS
    -> be_there, 1 AS snack FROM (Games INNER JOIN Players ON Games.homeTeam =
    -> Players.teamName) WHERE homeTeam IN (SELECT teamName FROM Players)
    -> UNION SELECT playerName, homeTeam AS against, gameTime, 0 AS snack
    -> FROM (Games INNER JOIN Players ON Games.awayTeam = Players.teamName)
    -> WHERE awayTeam IN (SELECT teamName FROM Players)) AS table1 WHERE
    -> playerName = 'Luca Pozzi' ORDER BY be_there;
+------------+-----------+---------------------+-------+
| playerName | against   | be_there            | snack |
+------------+-----------+---------------------+-------+
| Luca Pozzi | SpaceX    | 2019-09-07 08:00:00 |     1 |
| Luca Pozzi | Microsoft | 2019-09-07 09:00:00 |     1 |
| Luca Pozzi | Google    | 2019-09-07 10:00:00 |     1 |
| Luca Pozzi | NASA      | 2019-09-07 11:00:00 |     0 |
| Luca Pozzi | Intel     | 2019-09-07 12:00:00 |     0 |
+------------+-----------+---------------------+-------+
5 rows in set (0.00 sec)
```



```
mysql> SELECT playerName, against, gameTime AS be_there, snack, coachName
    -> AS lead_by FROM (SELECT Players.playerName, against, gameTime, snack,
    -> teamName FROM (SELECT * FROM (SELECT playerName, awayTeam AS
    -> against, gameTime, 1 AS snack FROM (Games INNER JOIN Players ON
    -> Games.homeTeam = Players.teamName) WHERE homeTeam IN (SELECT teamName
    -> FROM Players) UNION SELECT playerName, homeTeam AS against, gameTime,
    -> 0 AS snack FROM (Games INNER JOIN Players ON Games.awayTeam =
    -> Players.teamName) WHERE awayTeam IN (SELECT teamName FROM Players)) AS
    -> table1 WHERE playerName = 'Luca Pozzi' ORDER BY gameTime) AS table2
    -> INNER JOIN Players ON table2.playerName = Players.playerName) AS
    -> table3 INNER JOIN Teams ON table3.teamName = Teams.teamName;
+------------+-----------+---------------------+-------+-----------------+
| playerName | against   | be_there            | snack | lead_by         |
+------------+-----------+---------------------+-------+-----------------+
| Luca Pozzi | SpaceX    | 2019-09-07 08:00:00 |     1 | Travis Kalanick |
| Luca Pozzi | Microsoft | 2019-09-07 09:00:00 |     1 | Travis Kalanick |
| Luca Pozzi | Google    | 2019-09-07 10:00:00 |     1 | Travis Kalanick |
| Luca Pozzi | NASA      | 2019-09-07 11:00:00 |     0 | Travis Kalanick |
| Luca Pozzi | Intel     | 2019-09-07 12:00:00 |     0 | Travis Kalanick |
+------------+-----------+---------------------+-------+-----------------+
5 rows in set (0.01 sec)
```

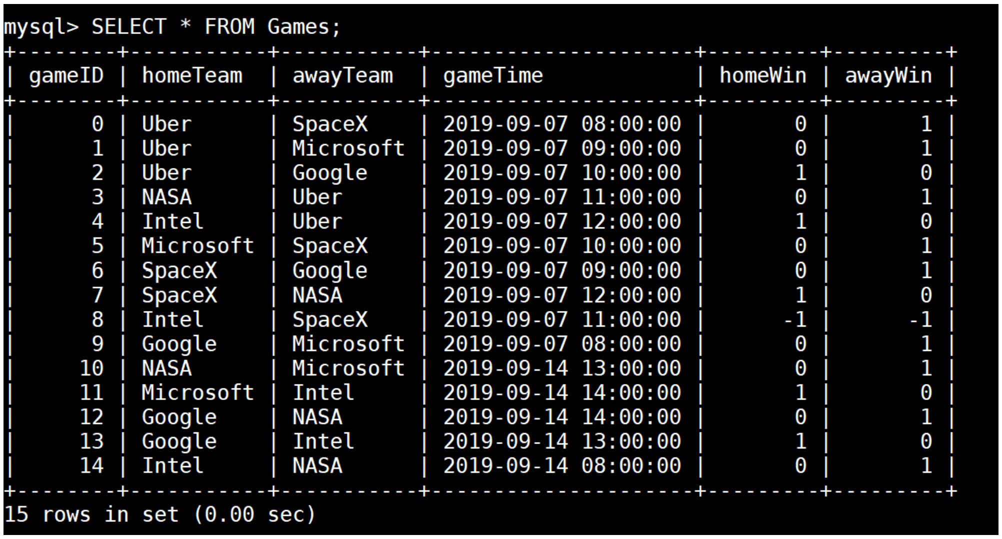
[5] *Extra credit if you can create a table that shows the win, tie, and loss record of each team in a single query (no stored partial results) on your original tables (Hint: you may want to use the UNION operator and sub-queries for this one.)*

To create this, first I updated the Games table with the results of the game to show whether the home team or the away team won or if it was a tie. The key for the results are: Win = 1, Loss = 0, and Tie = -1.



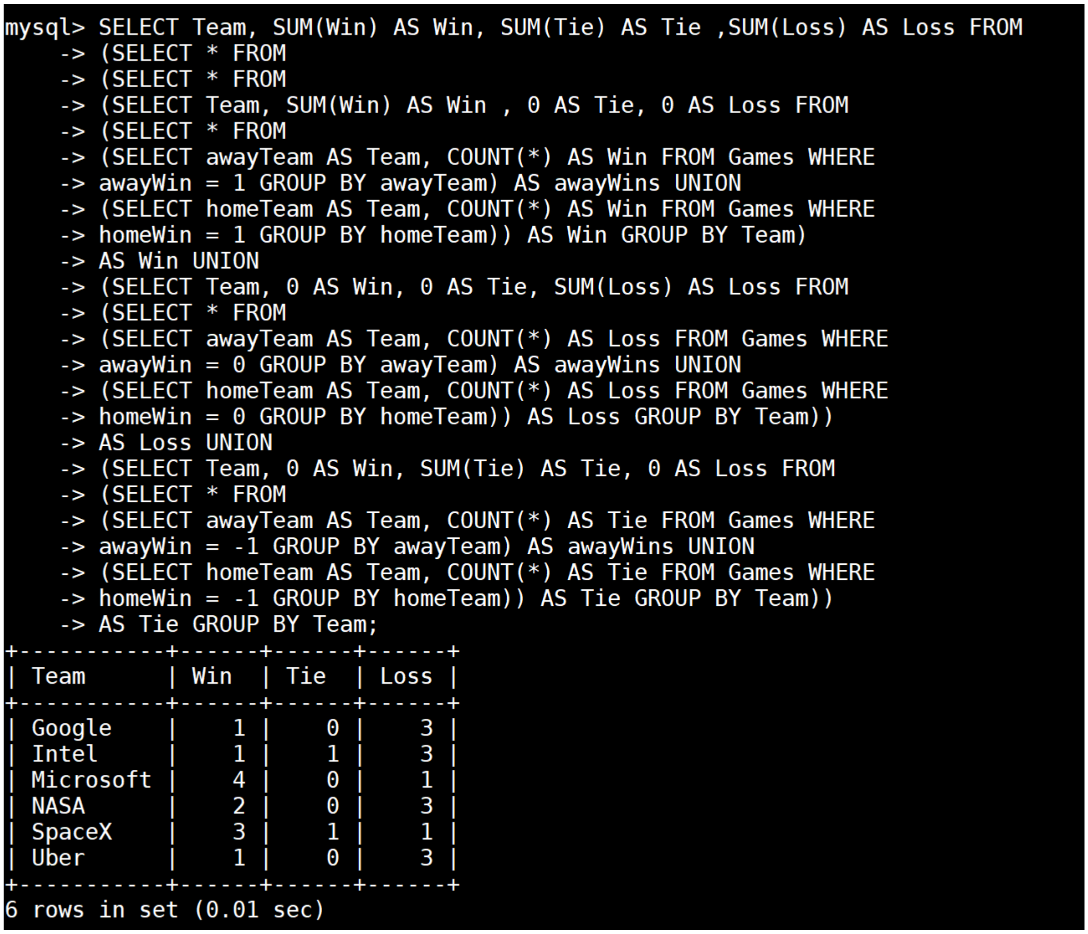
```
mysql> UPDATE Games SET homeWin = CASE
    -> WHEN gameID = 0 THEN 0
    -> WHEN gameID = 1 THEN 0
    -> WHEN gameID = 2 THEN 1
    -> WHEN gameID = 3 THEN 0
    -> WHEN gameID = 4 THEN 1
    -> WHEN gameID = 5 THEN 0
    -> WHEN gameID = 6 THEN 0
    -> WHEN gameID = 7 THEN 1
    -> WHEN gameID = 8 THEN -1
    -> WHEN gameID = 9 THEN 0
    -> WHEN gameID = 10 THEN 0
    -> WHEN gameID = 11 THEN 1
    -> WHEN gameID = 12 THEN 0
    -> WHEN gameID = 13 THEN 1
    -> WHEN gameID = 14 THEN 0
    -> END;
Query OK, 15 rows affected (0.01 sec)
Rows matched: 15  Changed: 15  Warnings: 0

mysql>
mysql> UPDATE Games SET awayWin = CASE
    -> WHEN gameID = 0 THEN 1
    -> WHEN gameID = 1 THEN 1
    -> WHEN gameID = 2 THEN 0
    -> WHEN gameID = 3 THEN 1
    -> WHEN gameID = 4 THEN 0
    -> WHEN gameID = 5 THEN 1
    -> WHEN gameID = 6 THEN 1
    -> WHEN gameID = 7 THEN 0
    -> WHEN gameID = 8 THEN -1
    -> WHEN gameID = 9 THEN 1
    -> WHEN gameID = 10 THEN 1
    -> WHEN gameID = 11 THEN 0
    -> WHEN gameID = 12 THEN 1
    -> WHEN gameID = 13 THEN 0
    -> WHEN gameID = 14 THEN 1
    -> END;
Query OK, 15 rows affected (0.01 sec)
Rows matched: 15  Changed: 15  Warnings: 0
```

The updated table is shown below.



```
mysql> SELECT * FROM Games;
+--------+-----------+-----------+---------------------+---------+---------+
| gameID | homeTeam  | awayTeam  | gameTime            | homeWin | awayWin |
+--------+-----------+-----------+---------------------+---------+---------+
|      0 | Uber      | SpaceX    | 2019-09-07 08:00:00 |       0 |       1 |
|      1 | Uber      | Microsoft | 2019-09-07 09:00:00 |       0 |       1 |
|      2 | Uber      | Google    | 2019-09-07 10:00:00 |       1 |       0 |
|      3 | NASA      | Uber      | 2019-09-07 11:00:00 |       0 |       1 |
|      4 | Intel     | Uber      | 2019-09-07 12:00:00 |       1 |       0 |
|      5 | Microsoft | SpaceX    | 2019-09-07 10:00:00 |       0 |       1 |
|      6 | SpaceX    | Google    | 2019-09-07 09:00:00 |       0 |       1 |
|      7 | SpaceX    | NASA      | 2019-09-07 12:00:00 |       1 |       0 |
|      8 | Intel     | SpaceX    | 2019-09-07 11:00:00 |      -1 |      -1 |
|      9 | Google    | Microsoft | 2019-09-07 08:00:00 |       0 |       1 |
|     10 | NASA      | Microsoft | 2019-09-14 13:00:00 |       0 |       1 |
|     11 | Microsoft | Intel     | 2019-09-14 14:00:00 |       1 |       0 |
|     12 | Google    | NASA      | 2019-09-14 14:00:00 |       0 |       1 |
|     13 | Google    | Intel     | 2019-09-14 13:00:00 |       1 |       0 |
|     14 | Intel     | NASA      | 2019-09-14 08:00:00 |       0 |       1 |
+--------+-----------+-----------+---------------------+---------+---------+
15 rows in set (0.00 sec)
```

After updating the table, I created the query to display the results of the teams.



```
mysql> SELECT Team, SUM(Win) AS Win, SUM(Tie) AS Tie ,SUM(Loss) AS Loss FROM
    -> (SELECT * FROM
    -> (SELECT * FROM
    -> (SELECT Team, SUM(Win) AS Win , 0 AS Tie, 0 AS Loss FROM
    -> (SELECT * FROM
    -> (SELECT awayTeam AS Team, COUNT(*) AS Win FROM Games WHERE
    -> awayWin = 1 GROUP BY awayTeam) AS awayWins UNION
    -> (SELECT homeTeam AS Team, COUNT(*) AS Win FROM Games WHERE
    -> homeWin = 1 GROUP BY homeTeam)) AS Win GROUP BY Team)
    -> AS Win UNION
    -> (SELECT Team, 0 AS Win, 0 AS Tie, SUM(Loss) AS Loss FROM
    -> (SELECT * FROM
    -> (SELECT awayTeam AS Team, COUNT(*) AS Loss FROM Games WHERE
    -> awayWin = 0 GROUP BY awayTeam) AS awayWins UNION
    -> (SELECT homeTeam AS Team, COUNT(*) AS Loss FROM Games WHERE
    -> homeWin = 0 GROUP BY homeTeam)) AS Loss GROUP BY Team))
    -> AS Loss UNION
    -> (SELECT Team, 0 AS Win, SUM(Tie) AS Tie, 0 AS Loss FROM
    -> (SELECT * FROM
    -> (SELECT awayTeam AS Team, COUNT(*) AS Tie FROM Games WHERE
    -> awayWin = -1 GROUP BY awayTeam) AS awayWins UNION
    -> (SELECT homeTeam AS Team, COUNT(*) AS Tie FROM Games WHERE
    -> homeWin = -1 GROUP BY homeTeam)) AS Tie GROUP BY Team))
    -> AS Tie GROUP BY Team;
+-----------+------+------+------+
| Team      | Win  | Tie  | Loss |
+-----------+------+------+------+
| Google    |    1 |    0 |    3 |
| Intel     |    1 |    1 |    3 |
| Microsoft |    4 |    0 |    1 |
| NASA      |    2 |    0 |    3 |
| SpaceX    |    3 |    1 |    1 |
| Uber      |    1 |    0 |    3 |
+-----------+------+------+------+
6 rows in set (0.01 sec)
```

Unfortunately, I cannot seem to figure out why my query is not working. It is removing a win count from teams Google and Uber.