**BAIS:3200 Database Management**

**NFL Project**

**Application Link:** <https://apex.oracle.com/pls/apex/r/nfl_data_project/nfl-data-analysis/home?session=3375488342123>

Kristina Gavin Bigsby

Group19 Ben Onderick, Scott, Anderson, Alexander Leitzke, Edward Kang, Dongzheng Su

**Introduction**

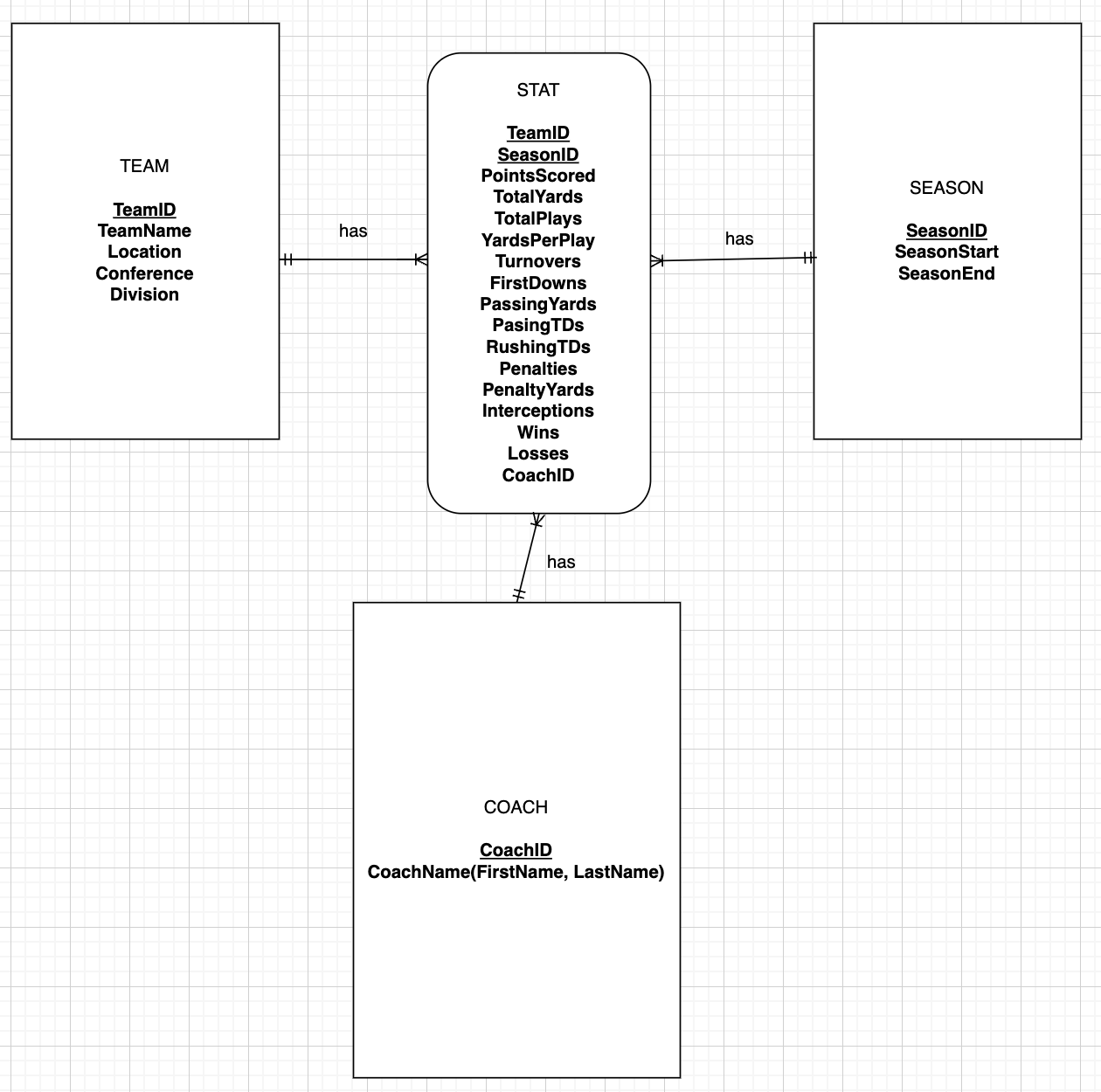
Football has been played since 1920 and has become one of America’s most popular sports. (<https://www.kaggle.com/datasets/ttalbitt/american-football-team-stats-1998-2019?resource=download&select=AmericanFootball98.csv>) In this dataset we will apply data analysis to identify the characteristics of a successful football team against non-successful teams. We will also examine trends throughout the decade of football we are analyzing and provide reasoning as to why these trends occur. Our database application may have many benefits to coaches, scouts, and analysts by outlining the important characteristics of professional football teams.

**Data**

The project data that we selected is a [Kaggle](https://www.kaggle.com/datasets/ttalbitt/american-football-team-stats-1998-2019?resource=download&select=AmericanFootball98.csv) data set that focuses on American football statistics from 1998 to 2019. The original data set contains each team’s single season statistics for each of the years, which ends up being 663 columns in total. There were about 60 different statistics that are shown for each team, including offensive, defensive and some opponent statistics. We decided to trim the data and use ten years of data, from 2009 – 2019. We have also decided to trim down the data to just include the main offensive statistics for both the team and the opponent team that they played. Each statistic is an average, except for the total amounts. In total, our final data set contains 15 columns, which describe the statistics for each team and 351 rows of data, which are the teams, and the season for which the statistics occur.

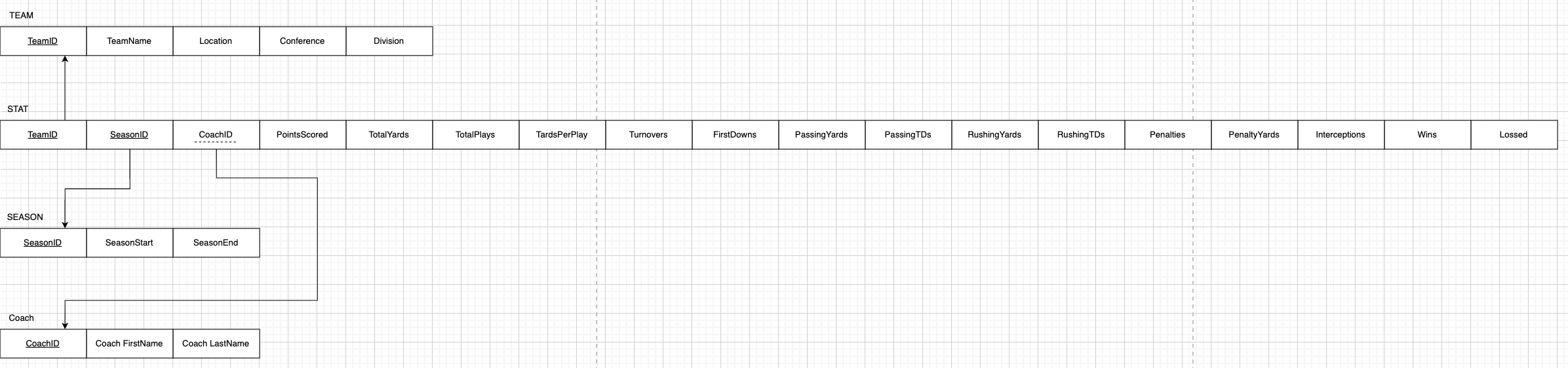
|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| TeamID | Text | Each team’s unique identifier |
| TeamName | Text | Each team’s name |
| SeasonID | Text | Each season’s unique identifier |
| SeasonStart | Date | Each season’s start year |
| SeasonEnd | Date | Each season’s end year |
| Location | Text | Each team’s location |
| Conference | Text | Each team’s conference |
| Division | Text | Each team’s division |
| Wins | Numeric | How many total wins in season |
| Loses | Numeric | How many total losses in season |
| PointsScored | Numeric | How many total points scored in season |
| TotalYards | Numeric | How many total yards gained in season |
| TotalPlays | Numeric | How many total plays ran in season |
| YardsPerPlay | Numeric | How many yards gained per play |
| FirstDowns | Numeric | How many total first downs gained in season |
| Turnovers | Numeric | How many total turnovers in season |
| PassingYards | Numeric | How many total yards passed for in season |
| PassingTDs | Numeric | How many total passing touchdowns scored in season |
| RushingYards | Numeric | How many total yards rushed for in season |
| RushingTDs | Numeric | How many total rush touchdowns scored in season |
| Penalties | Numeric | How many total penalties in season |
| PenaltyYards | Numeric | How many penalty yards given up in season |
| Interceptions | Numeric | How many interceptions throw in a season |
| FirstName | Text | Each teams’ head coach’s first name |
| LastName | Text | Each teams’ head coach’s last name |
| CoachID | Text | Unique identifier for each coach that has ever coached |

The primary entity in the database is TEAM, which is identified by TEAMID. The attributes for each team are treated as required because a team must have a name, location, conference, and division. STAT will be an associative table between TEAM, SEASON and COACH. The identifier of STAT is a composite of TeamID and SeasonID. We make the SEASON table as strong entity because the SeasonID is unique. The identifier CoachName in COACH Table is composite. Figure 1 displays the ERD for this data.



*Fig. 1 Entity relationship diagram (ERD)*

Based on this ERD, we normalized the data and created a relational schema with 4 tables. Figure 2 displays the graphical relational schema of the database. The TEAM table acts as the parent table. The SEASON, STAT and COACH tables as the child tables. For the STAT table, it has a surrogate key which is the combination of TeamID and SeasonID. For COACH Table, CoachID is the primary key.

*Fig. 2 Graphical relational schema*

**Database Implementation**

To implement the database in APEX, we wrote CREATE TABLE commands for each table in the relational schema.

*Team Table*

CREATE TABLE TEAM (

TeamID char(3) not null,

TeamName varchar(50) not null,

Location varchar(50) not null,

Conference char(3) not null,

Division varchar(10) not null,

CONSTRAINT TEAM\_PK PRIMARY KEY (TeamID)

);

*Season Table*

CREATE TABLE SEASON (

SeasonID char(4) not null,

SeasonStart date not null,

SeasonEnd date not null,

CONSTRAINT SEASON\_PK PRIMARY KEY (SeasonID)

);

*Coaches Table*

CREATE TABLE COACHES (

CoachID varchar(5) not null,

FirstName varchar(100) not null,

LastName varchar(100) not null,

CONSTRAINT COACHS\_PK PRIMARY KEY (CoachID)

);

*Stats Table*

CREATE TABLE STATS (

TeamID char(3) not null,

SeasonID char(4) not null,

PointsScored varchar(4) not null,

TotalYards varchar(5) not null,

TotalPlays varchar(5) not null,

YardsPerPlay varchar(5) not null,

Turnovers varchar(5) not null,

FirstDowns varchar(5) not null,

PassingYards varchar(5) not null,

PassingTDs varchar(5) not null,

RushingYards varchar(5) not null,

RushingTDs varchar(5) not null,

Penalties varchar(5) not null,

PenaltyYards varchar(5) not null,

Interceptions varchar(5) not null,

Wins number not null,

Losses number not null,

CoachID varchar(5) not null,

CONSTRAINT STATS\_PK PRIMARY KEY (TeamID,SeasonID)

CONSTRAINT COACHES\_PK FOREIGN KEY (CoachID) REFERENCES COACHES(CoachID)

);

**Analysis**

*Question 1*

Write a case query that returns teams from 2019 who have 14 or more wins as ‘Great’, 10 or more wins as ‘Good’, 7 or more wins as ‘Okay’, and if a team has less than 7 wins label them as ‘Poor’.

For question one, we wrote a case query for when a team wins 14 or more games they are classified as great. Teams with 10-13 wins were labeled as Good. 7-9 wins were Okay. And if a team had less, they were Poor. We ordered this Query to return based off wins in descending order.

SELECT TEAMNAME, WINS, CASE

WHEN WINS >= 14 THEN 'Great'

WHEN WINS >= 10 THEN 'Good'

WHEN WINS >= 7 THEN 'Okay'

ELSE 'Poor'

END AS SeasonPerformance

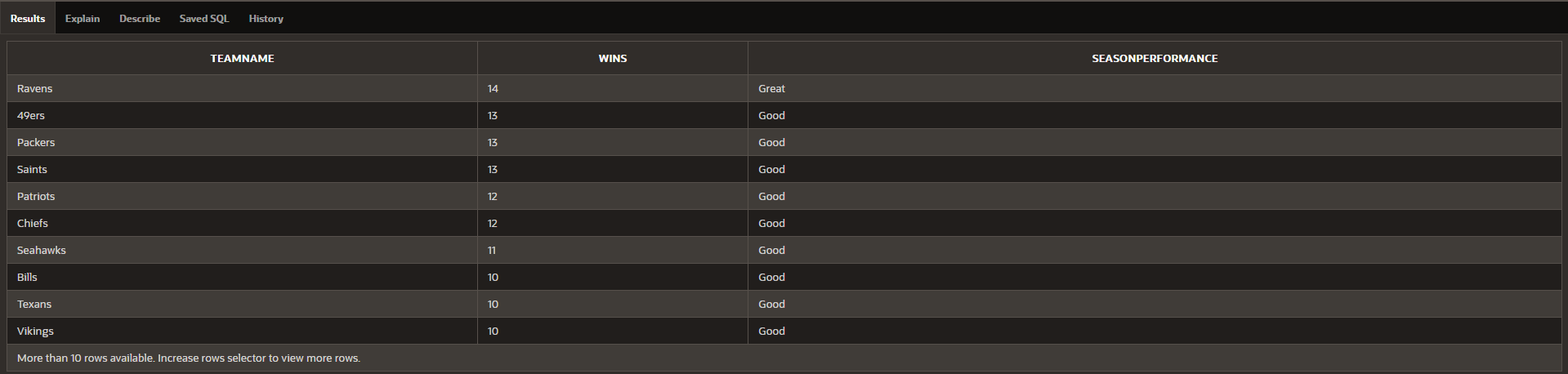
FROM TEAM

JOIN STATS ON TEAM.TEAMID = STATS.TEAMID JOIN SEASON ON STATS.SeasonID = SEASON.SeasonID

WHERE SEASON.SEASONID = '2019'

ORDER BY WINS DESC;

Below are the results of the query in *Fig. 3*: As shown in the returned table, the Ravens were the only team that was recognized as ‘Great’. Figure 3 returns all teams, only the top ten are shown below.

*Fig. 3*

*Question 2*

How many teams has every head coach worked for? What are their average wins throughout their coaching career?

For question two, we wrote a query that returns the head coaches full name which we used the CONCAT function for. We then count the number of teams that each coach has coached, along with the number of wins. Inside of the query we use a calculation with the number of wins and the number of losses pulled from WINS and LOSSES. All of this was pulled from the coach's table. For the order that we want the coach results in, we use ORDER BY in descending order, to return the coaches with the greatest number of wins at the top.

SELECT

CONCAT(CONCAT(FirstName, ' '), LastName) AS HEADCOACHNAME,

COUNT(DISTINCT TeamName) AS HowManyTeamsCoached,

SUM(Wins) AS TotalWins,

CONCAT(ROUND(SUM(WINS) / SUM(WINS + LOSSES) \* 100, 2), '%') AS WinPercentage,

ROUND(AVG(WINS),2) AS AverageWins

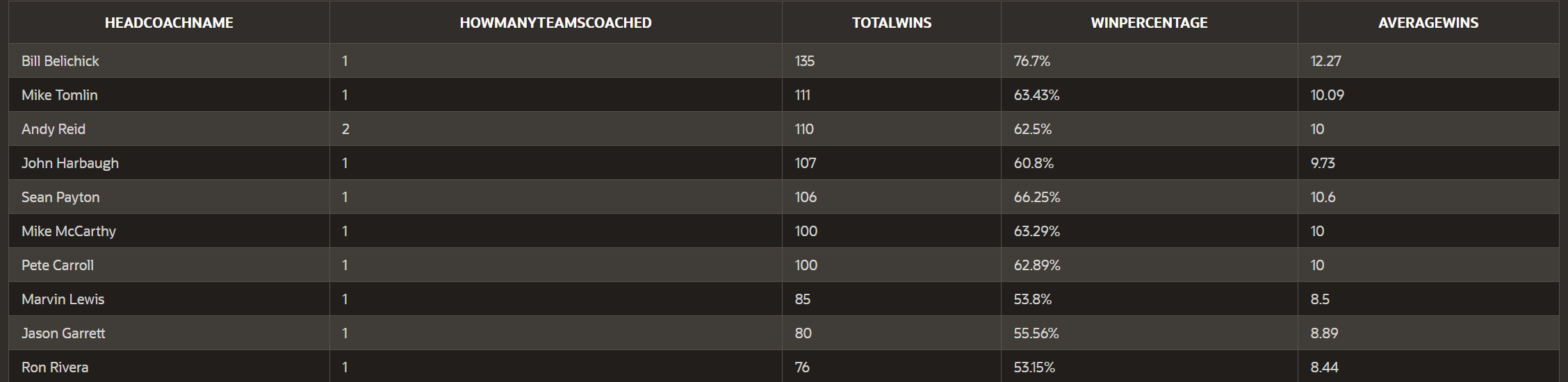
FROM COACHES

JOIN STATS ON COACHES.CoachID = STATS.CoachID JOIN TEAM ON STATS.TeamID = TEAM.TeamID

GROUP BY FirstName, LastName

ORDER BY TotalWins DESC;

Below are the results from our second query shown in *Fig. 4*. The results show 86 rows, only the first ten rows are shown below. Based on the results, Bill Belichick had the most amount of wins. As far as looking for the coach with the most teams coached, highest win percentage, and average wins, you would have to comb through all 86 rows of data which is not that efficient.

*Fig. 4*

Since that isn’t the most efficient way, all we had to do was change the ORDER BY statement. We were also interested in the coach with the highest win percentage. We wrote another query, that was the exact same but instead of TotalWins in the ORDER BY statement, we changed this to WinPercentage. Our query is below:

SELECT

CONCAT(CONCAT(FirstName, ' '), LastName) AS HEADCOACHNAME,

COUNT(DISTINCT TeamName) AS HowManyTeamsCoached,

SUM(Wins) AS TotalWins,

CONCAT(ROUND(SUM(WINS) / SUM(WINS + LOSSES) \* 100, 2), '%') AS WinPercentage,

ROUND(AVG(WINS),2) AS AverageWins

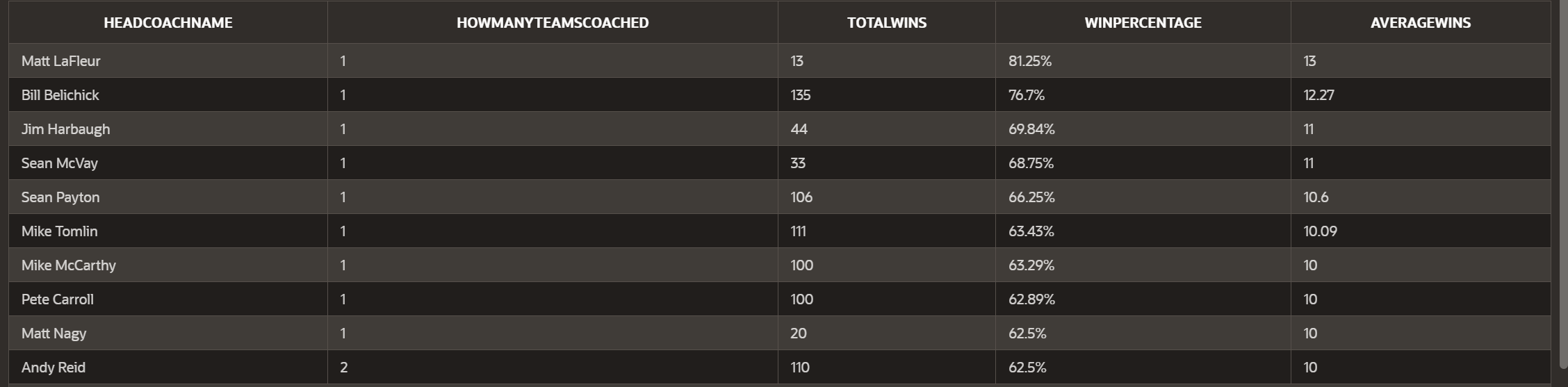
FROM COACHES

JOIN STATS ON COACHES.CoachID = STATS.CoachID JOIN TEAM ON STATS.TeamID = TEAM.TeamID

GROUP BY FirstName, LastName

ORDER BY WinPercentage DESC;

Next, we will show what the query returned in *Fig. 5*. Once again, the query returns 86 rows, but only the top ten coaches based off the highest win percentage are shown below. Matt LaFleur had the best win percentage over the 10 years, and Bill Belichick made another appearance, but this time ranked second. This data does not have a minimum number of games needed to be qualified, therefore the data could be skewed with the amount of total games coached.

*Fig. 5*

*Question 3*

What team has the most wins over the 10-year data span, how many points did they score in all 10 years? Return the top 10.

This query presents an analysis of the top 10 NFL teams based on the number of wins and total points scored. Our query utilizes a database with two tables, TEAM and STATS, to analyze the top 10 NFL teams based on the number of wins and total points scored.

The following SQL query was used to retrieve the top 10 teams based on their total wins and total points scored:

SELECT TeamName, SUM(WINS) AS TotalWins, SUM(POINTSSCORED) AS TotalPointsScored

FROM TEAM

JOIN STATS ON TEAM.TeamID = STATS.TeamID

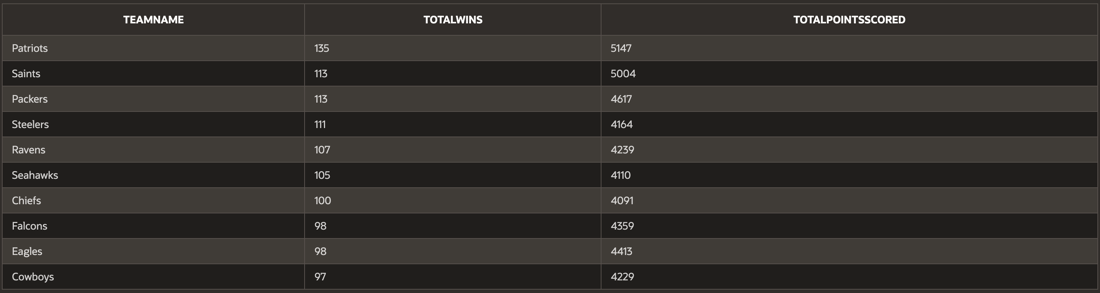
GROUP BY TeamName

ORDER BY TotalWins DESC

FETCH FIRST 10 ROWS ONLY;

The query joins the TEAM and STATS tables using the TeamID field, which serves as a common key between the two tables. The query then groups the data by TeamName and calculates the sum of WINS and POINTSSCORED for each team. The data is then ordered in descending order based on TotalWins and limited to the top 10 teams.

The top 10 NFL teams based on the number of wins and total points scored are as follows in *Fig. 6*:

*Fig. 6*

The results show that the New England Patriots have the highest number of total wins (135) and the highest total points scored (5147). This indicates that the Patriots are the top-performing team in the NFL during the specified period. The New Orleans Saints and the Green Bay Packers are tied in terms of total wins (113), but the Saints have scored more points (5004) than the Packers (4617), making the Saints the second-best team in the league.

*Question 4*

Who ran the ball for the most yards and touchdowns this decade? Who passed the ball for most yards and touchdowns?

This query analyzes NFL team performance in terms of rushing and passing yards during the period from 2010 to 2019. Data was obtained from a database containing relevant team and player statistics. The results provide insights into the teams that dominated in rushing and passing performance throughout the decade.

The database consists of several tables, including TEAM, STATS, and SEASON, which store information about teams, player statistics, and seasons, respectively. The following metrics were calculated:

1. Total rushing yards per team
2. Total rushing touchdowns per team
3. Total passing yards per team
4. Total passing touchdowns per team

*Rushing Yards & Touchdowns*

SELECT TeamName, SUM(RushingTDs) AS TotalRushingTDs, SUM(RushingYards) AS TotalRushingYards

FROM STATS

JOIN TEAM ON STATS.TeamID = TEAM.TeamID

GROUP BY TeamName

ORDER BY TotalRushingTDs DESC;

A screenshot of a computer

Description automatically generated with medium confidence*Fig. 7*

*Passing Yards & Touchdowns*

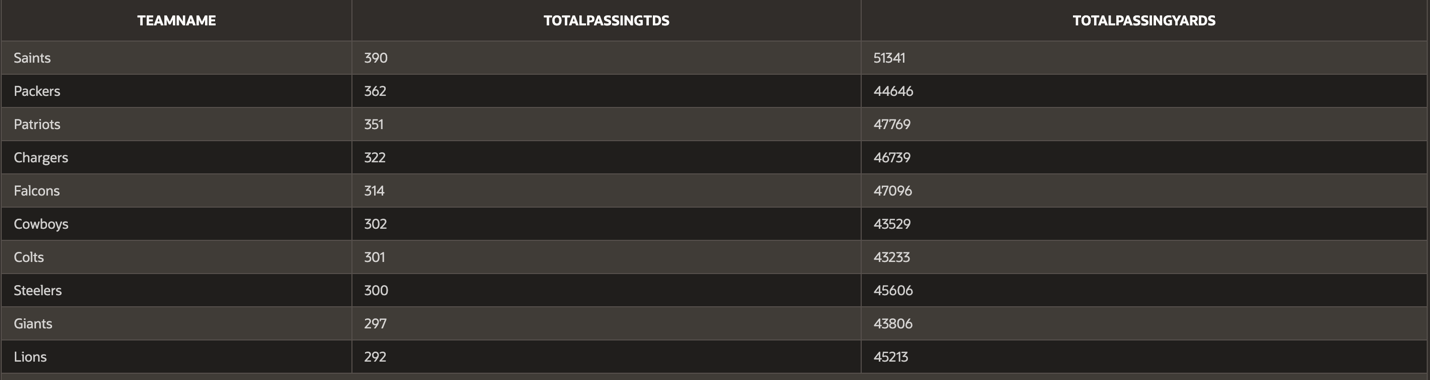
SELECT TeamName, SUM(PassingTDs) AS TotalPassingTDs, SUM(PassingYards) AS TotalPassingYards

FROM STATS

JOIN TEAM ON STATS.TeamID = TEAM.TeamID

GROUP BY TeamName

ORDER BY TotalPassingTDs DESC;

 *Fig. 8*

These two queries are basically the exact same thing, just change for the different statistics. They both select the team name from the TEAM table and the various rushing and passing stats from the STATS table. They then join to the STATS table to access the stats, and finally return all the teams stats for the decade.

*Question 5*

Which division had the most success over the 10 years? Consider wins as the most influential statistic.

For this question, we made a query that sums up the number of wins and labels them as TotalWins. Then groups division and conference while being ordered by wins in descending order (meaning the greatest number of wins will appear at the top).

SELECT Conference, Division, SUM(Wins) AS TotalWins

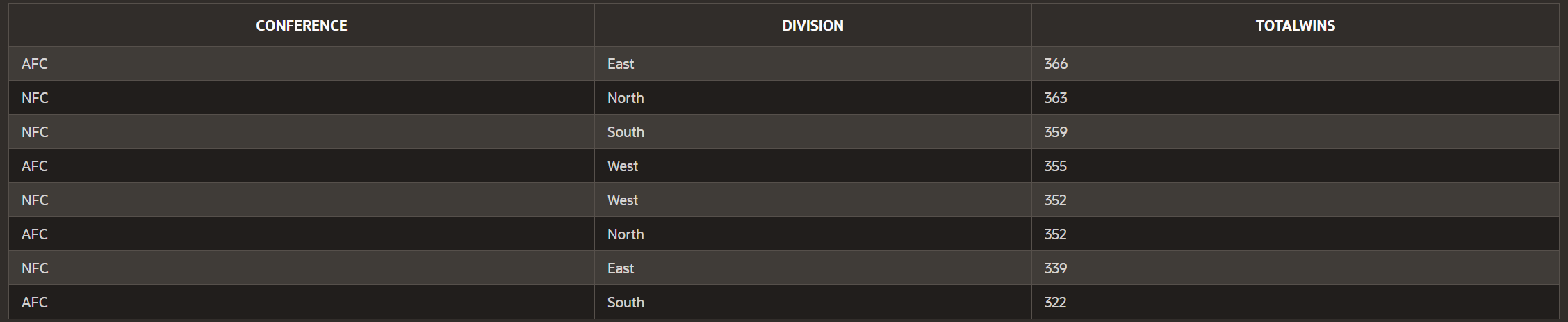
FROM TEAM

JOIN SEASON ON TEAM.TeamID = SEASON.TeamID

GROUP BY Division, Conference

ORDER BY TotalWins DESC;

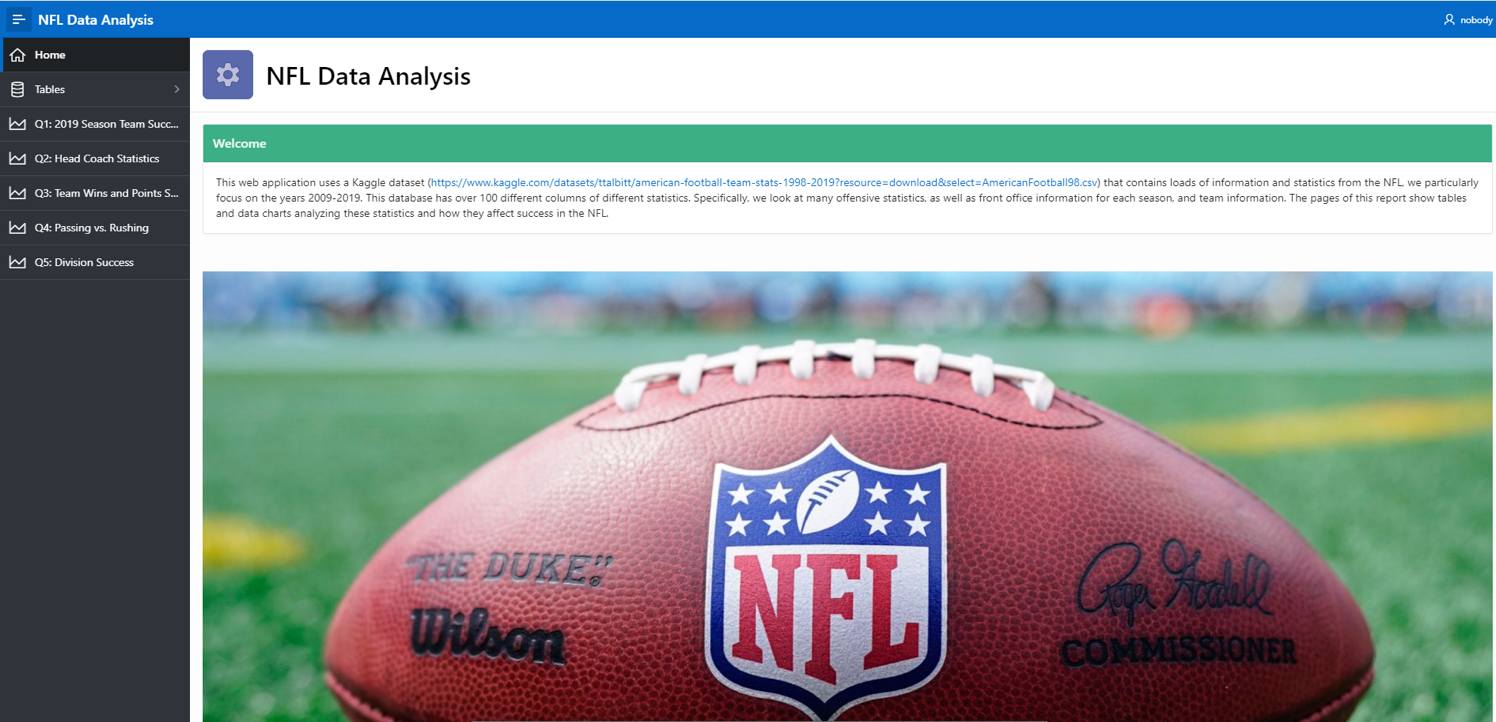
Below are the results from the query shown in *Fig. 9.* There are eight divisions with four in each conference (same division names in each conference). With the data shown below in *Fig. 9*. We observed that the AFC East was the most successful division in the NFL over the ten years solely based off wins.

*Fig. 9*

**Web Design**

*Home Page*

The home page of our web application explains what we will be doing for the project with our data hyperlinked. There is also a drop-down menu so the user can visit the other pages of our project. Our data was all about the NFL, so we figured there should be a football on the home page. Below in *Fig. 10* is our home page.

*Fig. 10*

*Tables*

For the next part of our application, we added a section to view our tables. This is to get a large view of our data and be able to view certain aspects at one time. They are all interactive reports, so they can be filtered and changed to the viewers request. We have 4 tables, a coaches table, team table, stats table and a season table. They are pictured below.

Table

Description automatically generated*Fig. 11*

Graphical user interface, table

Description automatically generated *Fig. 12*

Graphical user interface, table

Description automatically generated *Fig. 13*

Graphical user interface, application, table

Description automatically generated *Fig. 14*

*Queries*

Query 1: 2019 Season Team Success

On our query 1 page, we have a short description containing the question answered by the query, and short analysis of the results. Below the description is the full interactive report, as shown in *Fig. 15.* Users are able to search and sort by team name or season rating.

Graphical user interface, application

Description automatically generated *Fig. 15*

Query 2: Head Coach Statistics

The page for query 2 contains a short description, summarizing the question the query answers, and a short analysis. Below the short description is an interactive report, where users may sort, filter and search a coach by name, number of teams coached, average numbers of wins per season, total wins, and winning percentage. At the bottom of the page is a chart displaying a coach’s name, and their total number of wins across all seasons, as seen in *Fig. 16*.

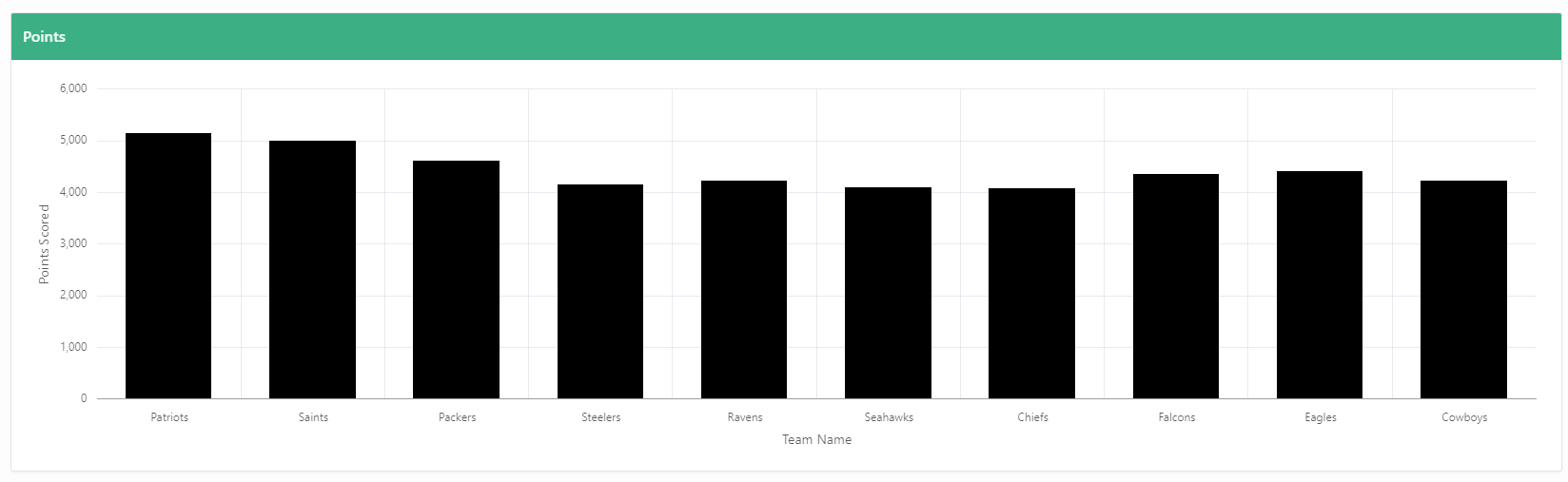
Chart

Description automatically generated*Fig. 16*

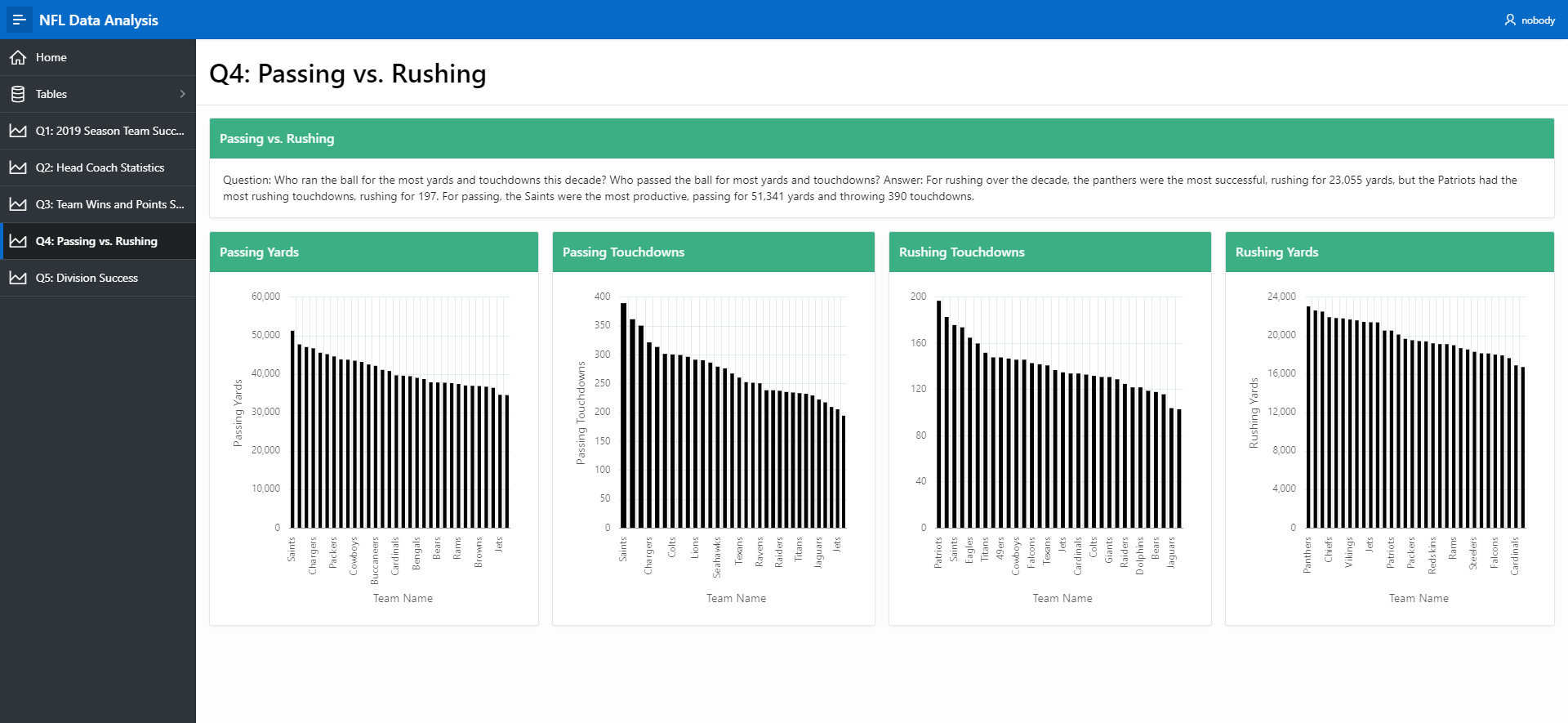
Query 3: Team Wins and Points Scored

On our query 3 page, a short description is displayed with the question answered by the query, and a short analysis. Below two graphs are displayed, the first one being total wins by team, and the second being total points scored by team, as show in Figure #.Graphical user interface, application, table

Description automatically generated *Fig. 17*

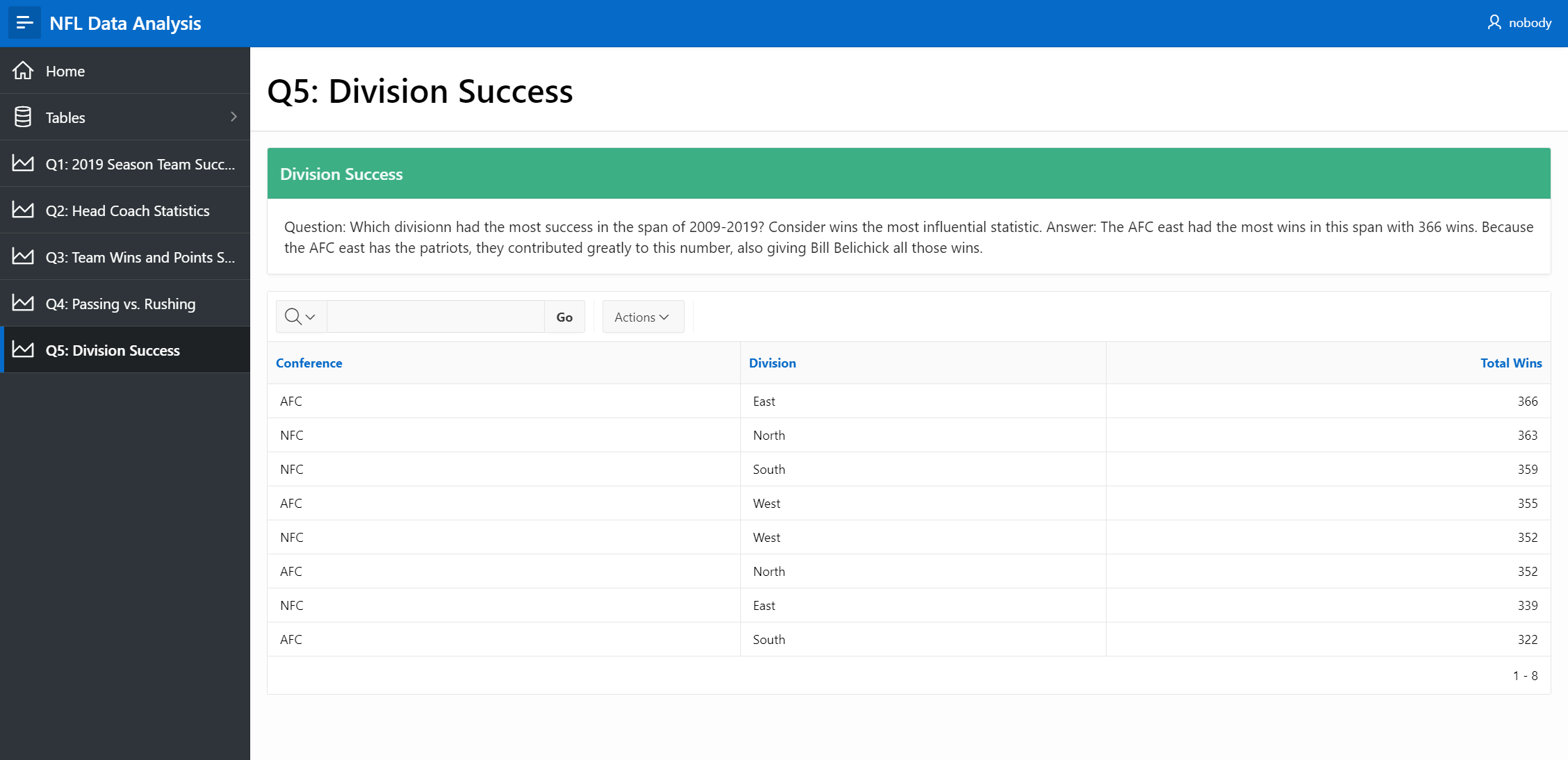
 *Fig. 18*

Query 4: Passing Td’s and yards along with Rushing Td’s and yards

On our query 4 page, we display the questions that we asked to get these results. There is also a short description of the findings from the dataset. All thirty-two teams are shown on the results, although some are not labeled unless you hover over their part of the charts. Below the charts are shown in *Fig. #* *Fig. 19*

Query 5: Division success based off wins

There is a description on the fifth query page, where the report is also interactive. The results are grouped by Conference and division.

*Fig. 20*