NFL Database Requirements

Josh Niemantsverdriet & Ethan Skowronski

Project Overview

This project aims to implement a relational database system for managing data related to the NFL (National Football League). The database will store and track information about teams, players, games, player statistics, stadiums, divisions, and conferences. This system will serve as a backend for applications such as team analysis tools, player performance dashboards, and more which will benefit both team staff and the media.

Stakeholders

- League staff: Require access to detailed information about games, teams, players, and performance statistics
- Team staff: Need tools to analyze player stats, game outcomes, and team matchups
- Sportscasters: Need clean data to present to fans and audiences
- Fans/Users: May interact through a front-end application to browse team details, player profiles, or view schedules
- Data Engineers: Require a clean, normalized backend to support analytics tools and visualizations

Data Requirements

Refer to relationship design & schema.

Use Cases

1. View Game Schedule
    a. Users can view upcoming games, including home/away teams, game time, and stadium location.
2. Analyze Player Performance
    a. Analysts can retrieve player stats across games or seasons, e.g., top 10 quarterbacks by passing yards.
3. Browse Team Information
    a. Users can explore teams, including their city, division, stadium, and players.

4. Track Division and Conference Standings
    a. Users can view how teams are grouped and compare performances across divisions or conferences.
5. Generate Reports
    a. Admins can create reports for YPA, touchdowns, sacks, and much more to determine the usefulness of players based on the stats releve to the players particular position on the team.
6. Game Hosting Management
    a. Stadium managers can query which games are scheduled at a specific stadium, including occupancy planning. This is particularly useful for teams like the Jets & Giants who share a stadium, or the Rams & Chargers who are in a similar situation.

Functional Requirements

- Query player stats filtered by game, team, or position, division, etc.
- Retrieve all players on a given team.
- Get all games a player has played in w/ statistics
- Return all games for a given team or pair of teams (useful for predicting outcomes)
- List all teams in a specific division and their records to predict playoff seeding

Non-Functional Requirements

- System should utilize Azure SQL database.
- Data must be normalized
- Foreign key constraints to maintain data integrity.

Assumptions

- Each player plays for one team at a time.
- Each game has exactly one home and one away team.
- Stats are only recorded if a player participates in at least one snap (play) of a game.

Conclusion

This NFL database system will provide a clear backend structure for sports analysis, team management, and user-facing applications. By implementing a clean schema and supporting core use cases, it will serve multiple stakeholders and support future expansion on any scale.