Design and Normalization Justification

We created a database for this project to explore the relationship between NFL player performance and overall team success across multiple seasons. We aimed to support accurate, flexible analysis by organizing data into clear, well-defined tables and minimizing redundancy. To achieve this, we built a schema that separates static information from dynamic, seasonal data.

The database includes six main tables: Player, Team, Season, PlayerTeamSeason, PlayerStats, and TeamStats. Each table has a specific purpose. The Player table holds player-specific information such as name and position details, which remain consistent regardless of season. Similarly, the Team table contains each NFL team's name and a unique ID. This separation ensures that descriptive data is not repeated across performance tables.

The Season table stores a single column for the year (2003, 2004, etc.), allowing all seasonal data to reference it. This helps maintain consistency throughout the database and enables comparisons across years.

We introduced the PlayerTeamSeason table to capture the many relationships between players, teams, and seasons. This join table records which team a player was affiliated with in a specific season. It prevents the need to duplicate team information in the PlayerStats table and supports accurate historical tracking, especially for players who change teams across seasons.

The PlayerStats table contains yearly player performance metrics, such as passing yards, rushing yards, receiving yards, touchdowns, interceptions, tackles, and field goals. Each record is linked to a player_id and season_year, allowing for precise tracking of a player's seasonal contributions.

The TeamStats table contains detailed team performance metrics for each season. These include traditional stats like wins, losses, and points scored, as well as advanced statistics like win/loss percentage, margin of victory (MOV), turnovers, yards gained, penalties, and expected points. The table links each record to a specific team and season, enabling high-level team performance analysis over time.

Key design decisions that guided our schema include:

● Using fact tables (PlayerStats, TeamStats) for seasonal performance metrics, separate from static identity tables like Player and Team.

● Avoiding data duplication by centralizing all descriptive data and using foreign keys instead of repeating values.

● Making PlayerTeamSeason to allow multi-table joins and historical player-team tracking without inflating the stats tables.

● Supporting analytical queries, such as:
○ Which positions contribute most to winning games?
○ How important is defence to winning games?
○ How important is a strong run/pass game?

This design makes the database both scalable and easy to maintain. It provides a reliable foundation for writing complex SQL queries and conducting meaningful statistical analysis.