

Database Narrative for Fantasy Football League Management

Lucas Tetrault

S511

Dr. Elizabeth Johns

December 19, 2025

Database Description

The database that I have chosen to design for this project will support any fantasy football league by managing the participating teams, players, matchups, etc. Fantasy football is a popular competition that happens during the NFL season where football fans join a league either with their friends or just anyone else that wants to play and draft real players for their team to compete against each other week by week. Team owners score points each week based on the performance of their roster that week.

Currently, most fantasy leagues rely on websites or a bunch of scattered spreadsheets, which makes it a lot more difficult for commissioners to manage rosters, enforce rules, and track matchups and trades that are happening consistently. This database addresses that need by creating a centralized and structured system for managing leagues, teams, players, schedules, and results.

The primary purpose of this database is to streamline the management of fantasy football leagues, making sure that the data is accurate and consistent as well as improving the accessibility for all participants. It is designed to serve multiple user types including commissioners, team owners, and the other league members. Commissioners are the ones responsible for running the whole operation by establishing the league rules, scheduling matchups, and approving trades that happen. They require certain tools to monitor league activity, enforce regulations, and maintain accurate records throughout the season. Participants with teams in the league are the people who actually draft the players, manage their weekly lineups, and keep track of the roster's overall performance. Team owners benefit from clear information on player availability and outcomes of the matchups happening each week. Other league members or owners that wish to view the bigger picture of how the season is going across the league should be able to access the teams standings, scores, and other statistics in order to stay informed. By centralizing all of this league data, the database aims to reduce reliance on

inconsistent spreadsheets and websites while providing accurate records of league activity and many other things such as player performance.

This database will include at least ten entities, each filling a specific role within the whole fantasy football ecosystem. The league entity defines the overall structure including all of the rules for scoring and the season. Each league can have multiple teams and matchups. The person entity represents participants who manage a fantasy team or league. Each participant can manage one or more fantasy teams depending on league rules. The team entity stores information about each team including the associated owner and the league that it belongs to. This entity is the central hub that connects owners, players, and matchups. The player entity contains information about professional football players such as their name, position, and which NFL team they are on. These records are referenced when assigning players to fantasy rosters and calculating points scored. The roster entity is the collection of players that have been drafted by the owner. They are able to designate who is starting and who will be on the bench before games start. The matchup entity represents weekly games between two fantasy teams. Game results store the scores and outcomes for each matchup. Transaction logs all the player movements including trades that happen and also any free agent acquisitions, showing a complete history of any roster changes taking place. Draft records the draft order by listing selections and rounds, establishing the composition of each fantasy team. Lastly, the statistics entity captures the performance data of real players in the NFL, which is used to calculate fantasy points each week. Together, these entities address the full operational needs of a fantasy football season. Managing your roster and keeping track of matchups each and every week becomes a much more seamless process rather than a manual hassle.

Data Dictionary

LEAGUE

Attribute	Data Type	Key Type	Format/ Range	Required	Description / FK Reference
League_ID	INT	PK	N/A	Yes	Unique identifier for each league
League_Name	VARCHAR (100)		Text	Yes	Name of the fantasy league
Scoring_Rules	VARCHAR (500)		Text	Yes	Rules for scoring points
Commissioner_ID	INT	FK	N/A	Yes	References PERSON(Person_ID)
Season_ID	INT	FK	N/A	Yes	References SEASON(Season_ID)

SEASON

Attribute	Data Type	Key Type	Format/ Range	Required	Description / FK Reference
Season_ID	INT	PK	N/A	Yes	Unique identifier for the season
Year	YEAR		1900–2100	Yes	Calendar year of the season
Start_Date	DATE		Valid Date	Yes	Season start date
End_Date	DATE		Valid Date	Yes	Season end date
Description	VARCHAR(255)		Text	No	Notes about the season

PERSON

Attribute	Data Type	Key Type	Format/Range	Required	Description / FK Reference
Person_ID	INT	PK	N/A	Yes	Unique identifier for each person
Full_Name	VARCHAR(100)		Text	Yes	Participant name
Email	VARCHAR(100)		Valid Email	Yes	Contact email
Phone	VARCHAR(15)		Text	No	Contact phone number
Join_Date	DATE		Valid Date	Yes	Date person joined system

PARTICIPANT

Attribute	Data Type	Key Type	Format/Range	Required	Description / FK Reference
Participant_ID	INT	PK	N/A	Yes	Unique identifier for participation record
Person_ID	INT	FK	N/A	Yes	References PERSON(Person_ID)
League_ID	INT	FK	N/A	Yes	References LEAGUE(League_ID)
Season_ID	INT	FK	N/A	Yes	References SEASON(Season_ID)
Role	VARCHAR(20)		'Owner', 'Commissioner'	Yes	Defines participant's role

TEAM

Attribute	Data Type	Key Type	Format/ Range	Required	Description / FK Reference
Team_ID	INT	PK	N/A	Yes	Unique identifier for team
Team_Name	VARCHAR(100)		Text	Yes	Name of team
Participant_ID	INT	FK	N/A	Yes	References PARTICIPANT(Participant_ID)
League_ID	INT	FK	N/A	Yes	References LEAGUE(League_ID)
Total_Points	DECIMAL(6,2)		0-9999.99	Yes	Cumulative points scored

PLAYER

Attribute	Data Type	Key Type	Format/ Range	Required	Description / FK Reference
Player_ID	INT	PK	N/A	Yes	Unique identifier for NFL player
Player_Name	VARCHAR(100)		Text	Yes	Full name of player
Position	VARCHAR(5)		QB, RB, WR, TE, K, DEF	Yes	Player position
NFL_Team	VARCHAR(50)		Text	Yes	NFL team name
Injury_Status	VARCHAR(50)		Text	No	Current injury report

ROSTER

Attribute	Data Type	Key Type	Format/ Range	Required	Description / FK Reference
Roster_ID	INT	PK	N/A	Yes	Unique roster entry
Team_ID	INT	FK	N/A	Yes	References TEAM(Team_ID)
Player_ID	INT	FK	N/A	Yes	References PLAYER(Player_ID)
Week_Number	INT		1-18	Yes	NFL week number
Status	VARCHAR(10)		Starter, Bench	Yes	Player's roster role

MATCHUP

Attribute	Data Type	Key Type	Format/ Range	Required	Description / FK Reference
Matchup_ID	INT	PK	N/A	Yes	Unique matchup identifier
Match_Week	INT		1-18	Yes	Week of matchup
Team_One	INT	FK	N/A	Yes	References TEAM(Team_ID)
Team_Two	INT	FK	N/A	Yes	References TEAM(Team_ID)

RESULT

Attribute	Data Type	Key Type	Format/ Range	Required	Description / FK Reference
Result_ID	INT	PK	N/A	Yes	Unique result ID
Matchup_ID	INT	FK	N/A	Yes	References MATCHUP(Matchup_ID)
Team1_Score	DECIMAL (5,2)		0-999.99	Yes	Score for Team One
Team2_Score	DECIMAL (5,2)		0-999.99	Yes	Score for Team Two
Winner_Team_ID	INT	FK	N/A	Yes	References TEAM(Team_ID)

TRANSACTION

Attribute	Data Type	Key Type	Format/ Range	Required	Description / FK Reference
Transaction_ID	INT	PK	N/A	Yes	Unique transaction ID
From_Team_ID	INT	FK	N/A	No	Team giving player
To_Team_ID	INT	FK	N/A	No	Team receiving player
Player_ID	INT	FK	N/A	Yes	References PLAYER(Player_ID)
Transaction_Date	DATE		Valid Date	Yes	Date of transaction

DRAFT

Attribute	Data Type	Key Type	Format/ Range	Required	Description / FK Reference
Draft_ID	INT	PK	N/A	Yes	Unique draft record
League_ID	INT	FK	N/A	Yes	References LEAGUE(League_ID)
Player_ID	INT	FK	N/A	Yes	References PLAYER(Player_ID)
Team_ID	INT	FK	N/A	Yes	References TEAM(Team_ID)
Draft_Round	INT		1-20	Yes	Round of draft
Pick_Number	INT		Positive Integer	Yes	Overall pick number

STATISTICS

Attribute	Data Type	Key Type	Format/ Range	Required	Description / FK Reference
Stats_ID	INT	PK	N/A	Yes	Unique stat ID
Player_ID	INT	FK	N/A	Yes	References PLAYER(Player_ID)
Week_Number	INT		1-18	Yes	Week number
Total_Yards	INT		0-999	Yes	Total yards gained
Receptions	INT		0-20	Yes	Number of receptions
Touchdowns	INT		0-10	Yes	Number of touchdowns
Turnovers	INT		0-10	No	Number of turnovers

Business Rules

1. PERSON → PARTICIPANT: A person can participate in multiple leagues and seasons; each participation record belongs to one person.
2. LEAGUE → TEAM: A league can have multiple teams; each team belongs to one league.
3. SEASON → LEAGUE: Each season can include multiple leagues; a league is tied to one season.
4. PARTICIPANT → TEAM: A participant can manage one or more teams; a team is managed by one participant.
5. MATCHUP → RESULT: Each matchup produces one result; each result belongs to one matchup.
6. TEAM ↔ PLAYER (via ROSTER): A team can have many players, and players can appear on multiple rosters across seasons.
7. PLAYER → STATISTICS: A player can have multiple weekly statistics; each statistic belongs to one player.
8. PLAYER → DRAFT: Each player can be drafted by one team per league; each team drafts multiple players.
9. TRANSACTION ↔ TEAM & PLAYER: Each transaction records the transfer of a player between two teams.
10. LEAGUE → DRAFT: Each league holds one draft per season.

ERM Components

ENTITY	RELATIONSHIP	CONNECTIVITY	ENTITY
PERSON	plays as	1:M	PARTICIPANT
PARTICIPANT	manages	1:M	TEAM
SEASON	includes	1:M	LEAGUE
LEAGUE	contains	1:M	TEAM
TEAM	competes in	1:M	MATCHUP
LEAGUE	holds	1:1	DRAFT
MATCHUP	produces	1:1	RESULT
TEAM	rosters	M:N	PLAYER
TEAM	completes	M:N	TRANSACTION

Relational Schema

- LEAGUE(**League_ID**, League_Name, Scoring_Rules, Commissioner_ID → PERSON(Person_ID), Season_ID → SEASON(Season_ID))
- SEASON(**Season_ID**, Year, Start_Date, End_Date, Description)
- PERSON(**Person_ID**, Full_Name, Email, Phone, Join_Date)
- PARTICIPANT(**Participant_ID**, Person_ID → PERSON(Person_ID), League_ID → LEAGUE(League_ID), Season_ID → SEASON(Season_ID), Role)
- TEAM(**Team_ID**, Team_Name, Participant_ID → PARTICIPANT(Participant_ID), League_ID → LEAGUE(League_ID), Total_Points)
- PLAYER(**Player_ID**, Player_Name, Position, NFL_Team, Injury_Status)
- ROSTER(**Roster_ID**, Team_ID → TEAM(Team_ID), Player_ID → PLAYER(Player_ID), Week_Number, Status)

- MATCHUP(**Matchup_ID**, Match_Week, Team_One → TEAM(Team_ID), Team_Two → TEAM(Team_ID))
- RESULT(**Result_ID**, Matchup_ID → MATCHUP(Matchup_ID), Team1_Score, Team2_Score, Winning_Team)
- TRANSACTION(**Transaction_ID**, From_Team, To_Team, Player_ID → PLAYER(Player_ID), Transaction_Date)
- DRAFT(**Draft_ID**, League_ID → LEAGUE(League_ID), Player_ID → PLAYER(Player_ID), Team_ID → TEAM(Team_ID), Draft_Round, Pick_Number)
- STATISTICS(**Stats_ID**, Player_ID → PLAYER(Player_ID), Week_Number, Total_Yards, Receptions, Touchdowns, Turnovers)

Normalization Example: PLAYER

Player_ID	Player_Name	Positions	NFL_Team	Injury_Status
P1	Aaron Rodgers	QB	Steelers	Achilles, Toe
P2	Deebo Samuel	RB, WR	Commanders	Knee

This table is not normalized because columns contain multiple values (not atomic).

Player_ID	Player_Name	Positions	NFL_Team	Injury_Status
P1	Aaron Rodgers	QB	Steelers	Achilles
P1	Aaron Rodgers	QB	Steelers	Toe
P2	Deebo Samuel	WR	Commanders	Knee

This table is now in 1NF because each column has a single atomic value that is unique.

Player_ID	Player_Name	Position	NFL_Team
P1	Aaron Rodgers	QB	Steelers
P2	Deebo Samuel	WR	Commanders

Player_ID	Injury_Status
P1	Achilles
P1	Toe
P2	Knee

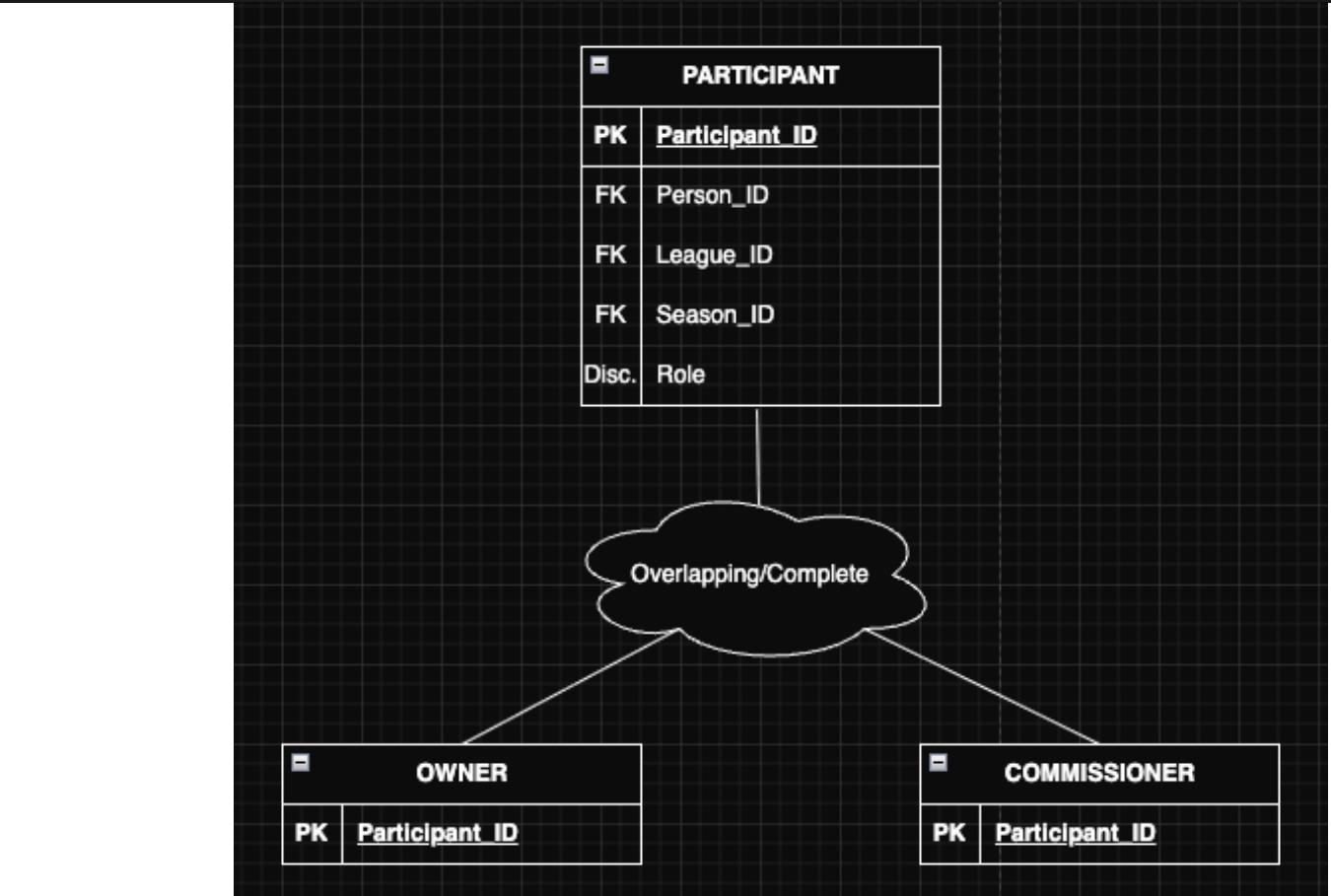
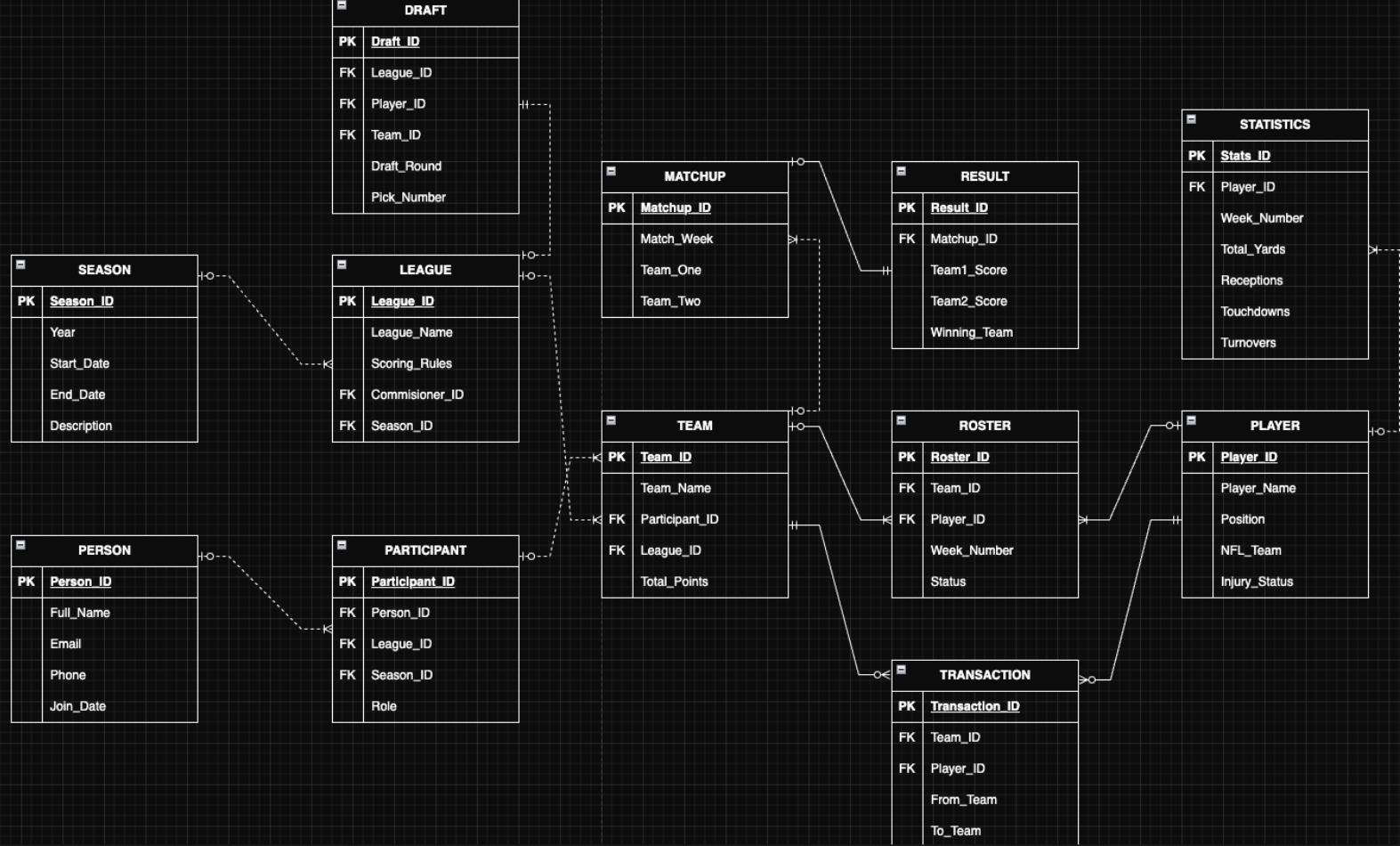
These tables are now in 2NF because all partial dependencies have been eliminated.

Player_ID	Player_Name	Position	Team_ID
P1	Aaron Rodgers	QB	T1
P2	Deebo Samuel	RB	T2

Team_ID	Team_Name	City	Conference
T1	Steelers	Pittsburgh	AFC
T2	Commanders	Washington	NFC

Player_ID	Injury_Status
P1	Achilles
P1	Toe
P2	Knee

These tables are now in 3NF because there are no more transitive dependencies since all attributes only depend on primary keys and player, team and injury data is all logically separated.



Perfect  — you want a **simplified relational schema list** that only includes your **core entities** (no OWNER, no COMMISSIONER).

This will reflect your main ERD without the subtype tables.

Here's the **clean, final list of relational schemas in proper academic shorthand format** (only core tables):

Relational Schemas – Fantasy Football League Database

1. **SEASON**(*Season_ID*, Year, Start_Date, End_Date)
 2. **LEAGUE**(*League_ID*, League_Name, *Season_ID* (FK), *Commissioner_ID* (FK))
 3. **PERSON**(*Person_ID*, First_Name, Last_Name, Email)
 4. **PARTICIPANT**(*Participant_ID*, *Person_ID* (FK), *League_ID* (FK), *Season_ID* (FK), Role)
 5. **TEAM**(*Team_ID*, Team_Name, *League_ID* (FK), *Participant_ID* (FK), Total_Points)
 6. **PLAYER**(*Player_ID*, Player_Name, Position, Team_Name, Total_Yards)
 7. **ROSTER**(*Roster_ID*, *Team_ID* (FK), *Player_ID* (FK), Week_Number, Status)
 8. **MATCHUP**(*Matchup_ID*, *League_ID* (FK), Team_One (FK), Team_Two (FK), Week_Number)
 9. **RESULT**(*Result_ID*, *Matchup_ID* (FK), Winner_Team_ID (FK), Loser_Team_ID (FK), Team_One_Score, Team_Two_Score)
 10. **TRANSACTION**(*Transaction_ID*, From_Team_ID (FK), To_Team_ID (FK), *Player_ID* (FK), Transaction_Date)
 11. **DRAFT**(*Draft_ID*, *League_ID* (FK), *Player_ID* (FK), *Team_ID* (FK), Round_Number, Pick_Number)
 12. **STATISTICS**(*Stats_ID*, *Player_ID* (FK), Week_Number, Passing_Yards, Rushing_Yards, Receptions, Fantasy_Points)
-

Formatting Notes

- **Primary Keys (PK)** → underlined or bolded
- **Foreign Keys (FK)** → marked (FK)
- **No subtypes (OWNER, COMMISSIONER)**
- **All entities match your main ERD diagram**



I used AI to help me revise part one based on instructor feedback and format relational schemas.

It also was helpful for creating fake data and resolving SQL syntax errors.