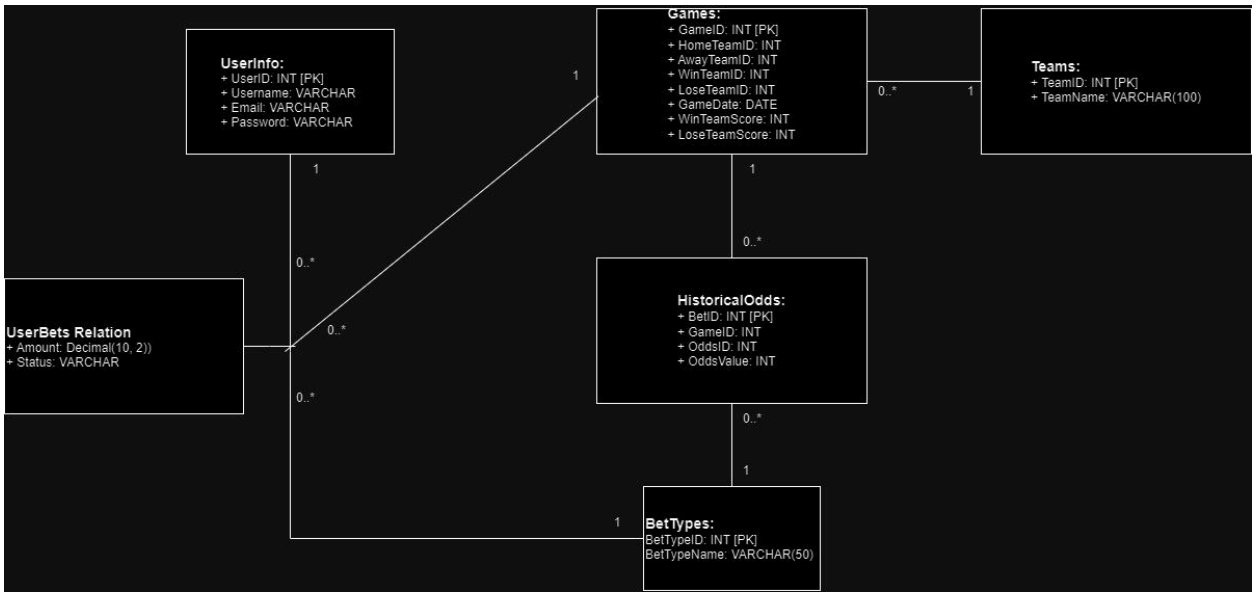
UML Diagram:



1. UserInfo

- Assumptions:
    - Each user must have a unique ID (UserID) to distinguish them from others.
    - Username, email, and password are essential for user login and authentication.
- Why Entity:
    - The UserInfo table is an entity because it represents a core actor in the system, and its attributes are user-specific data.
    - It's more efficient to keep this information in a separate table rather than embedding it in other tables like UserBets, as it would violate normalization rules and cause redundancy.
- Cardinality:
    - A single user can have multiple bets however, each bet record corresponds to one user - (1 to N relationship between UserInfo and UserBets).

UserBets Entity (3-Way Relation Table)

- Assumptions:
    - Each bet is identified by a composite key: UserID, GameID, and BetTypeID. These three columns will be primary keys for UserBets and uniquely define a bet.
    - Amount represents the wager placed by the user.
    - Status tracks the outcome of the bet (e.g., won, lost, pending).
- Why Entity:
    - UserBets serves as a relation table connecting UserInfo, Games, and BetTypes. It allows a user to place a specific type of bet on a specific game.
    - Storing betting information within UserInfo would lead to redundancy, as it would duplicate user information for each bet.

- Cardinality:
    - A single user can place multiple bets, with each bet related to a specific game and type (N to 1 with UserInfo, Games, and BetTypes).

3. Games Entity

- Assumptions:
    - A GameID uniquely identifies each game.
    - HomeTeamID and AwayTeamID reference the teams playing, and WinTeamID captures the result.
    - GameDate records when the game took place.
- Why Entity:
    - The Games table is a separate entity because it stores unique games' information and allows easy tracking of historical data.
    - Storing game information as an attribute within UserBets would lead to redundancy, as multiple users may place bets on the same game.
- Cardinality:
    - A game can have multiple bets associated with it (1 to N relationship between Games and UserBets).
    - A team can be involved in many games(1 to N relationships between Teams and Games).

4. HistoricalOdds Entity

- Assumptions:
    - OddsID references different odds types (e.g., moneyline, spread).
    - OddsValue stores the numerical value of odds for a particular game.
- Why Entity:
    - HistoricalOdds is modeled as an entity to store specific betting odds for each game and bet type. This allows easy updates and queries for the odds related to past games.
    - Storing odds data directly within the Games or UserBets table would cause data duplication and inefficiencies.
- Cardinality:
    - Each odds record references a single game and a single bet type (N to 1 relationships with Games and BetTypes).

5. BetTypes Entity

- Assumptions:
    - Contains distinct types of bets that users can place (e.g., win/loss, over/under).
    - BetTypeName provides a description of the type.
- Why Entity:
    - Bet types are abstracted into a separate entity to avoid data duplication and ensure consistency. It allows the program to add or update bet types without altering the bet records.

- Cardinality:
    - A bet type can be applied to multiple bets (1 to N relationship between BetTypes and UserBets).

6. Teams Entity

- Assumptions:
    - Each team is uniquely identified by TeamID.
    - TeamName stores the name of the team.
- Why Entity:
    - Teams are abstracted as separate entity because each team can participate in many games, and each game involves two teams (home and away).
    - Storing team information directly within Games would lead to redundancy and would not allow efficient updates.
- Cardinality:
    - A single team can participate in many games (1 to N relationship between Teams and Games).

Normalization:

Step-by-Step BCNF for each entity:
1. UserInfo

- FDs:
    - UserID → Username, Email, Password
- BCNF Verification:
    - This table is in BCNF because UserID, the primary key, uniquely determines all other attributes. There are no partial or transitive dependencies.

2. UserBets

Step 1: List All FDs

- FD1: BetID → UserID, GameID, BetTypeID, Amount, Status
- FD2: (UserID, GameID, BetTypeID) → Amount, Status

Step 2: Check if in BCNF

- FD1: BetID → UserID, GameID, BetTypeID, Amount, Status
    - This would satisfy BCNF if BetID were the primary key.
- FD2: (UserID, GameID, BetTypeID) → Amount, Status
    - This FD violates BCNF since (UserID, GameID, BetTypeID) is not a superkey.

Step 3: Resolve BCNF Violation by Decomposition

1. Pick Violation FD: f: (UserID, GameID, BetTypeID) → Amount, Status
2. Compute Closure (A+): (UserID, GameID, BetTypeID)+ = {UserID, GameID, BetTypeID, Amount, Status}

3. Create R1 and R2:
    - R1 = (UserID, GameID, BetTypeID, Amount, Status)
    - R2 = Since there are no additional attributes, UserBets becomes the 3-way relation table with a composite primary key: (UserID, GameID, BetTypeID).
4. After decomposition, UserBets is a 3-way relation table satisfying BCNF.

3. Games

- FDs:
    - GameID → HomeTeamID, AwayTeamID, WinTeamID, LoseTeamID, GameDate, WinTeamScore, LoseTeamScore
- BCNF Verification:
    - GameID, the primary key, uniquely determines all other attributes in the table. There are no partial or transitive dependencies.

4. HistoricalOdds

- FDs:
    - (BetID, BetTypeID) → GameID, OddsValue
- BCNF Verification:
    - The composite primary key (BetID, BetTypeID) determines GameID and OddsValue. There are no partial or transitive dependencies.

5. BetTypes

- Functional Dependencies (FDs):
    - BetTypeID → BetTypeName
- BCNF Verification:
    - The primary key BetTypeID uniquely determines BetTypeName. There are no partial or transitive dependencies.

6. Teams

- Functional Dependencies (FDs):
    - TeamID → TeamName
- BCNF Verification:
    - TeamID, the primary key, uniquely determines TeamName. There are no partial or transitive dependencies.

We used BCNF to normalize to eliminate more redundancy and provide better data integrity.

Relational Schema:

UserInfo Table

- UserID: INT [PK]
- Username: VARCHAR
- Email: VARCHAR
- Password: VARCHAR

Games Table

- GameID: INT [PK]
- HomeTeamID: INT
- AwayTeamID: INT
- WinTeamID: INT
- LoseTeamID: INT
- GameDate: DATE
- WinTeamScore: INT
- LoseTeamScore: INT

Teams Table

- TeamID: INT [PK]
- TeamName: VARCHAR(100)

UserBets Table (3-Way Relation)

- UserID: INT [PK]
- GameID: INT [PK]
- BetTypeID: INT [PK]
- Amount: DECIMAL(10, 2)
- Status: VARCHAR

BetTypes Table

- BetTypeID: INT [PK]
- BetTypeName: VARCHAR(50)

HistoricalOdds Table

- BetID: INT [PK]
- GameID: INT
- BetTypeID: INT
- OddsValue: DECIMAL(5, 2)