**Gamblers-Anonymous Database Design – Dakota List**

**Database Name:** GamblersAnonymous

**Purpose:** To store and manage data for a poker game application.

**Normalization Level:** Third Normal Form (3NF)

**Tables:**

1. **Players Table:** Primary Key: playerID Columns:
   * playerName: VARCHAR(255)
   * playerBlacklist: BOOLEAN
   * avatar: VARCHAR(255)
   * virtualCurrencyBalance: DECIMAL(10,2)
   * playerStatus: ENUM('active', 'folded', 'waiting')
   * playerRank: INTEGER
2. **Games Table:** Primary Key: gameID Columns:
   * roundNumber: INTEGER
   * limitType: ENUM('limit', 'no-limit', 'pot-limit')
   * smallBlind: DECIMAL(10,2)
   * bigBlind: DECIMAL(10,2)
   * currentPot: DECIMAL(10,2)
   * dealerPosition: INTEGER
   * playerID: INTEGER (Foreign key referencing Players table)
3. **CommunityCards Table:** Primary Key: communityCardsID Columns:
   * gameID: INTEGER (Foreign key referencing Games table)
   * card1: VARCHAR(255)
   * card2: VARCHAR(255)
   * card3: VARCHAR(255)
   * card4: VARCHAR(255)
   * card5: VARCHAR(255)
4. **PlayerCards Table:** Primary Key: playerCardsID Columns:
   * gameID: INTEGER (Foreign key referencing Games table)
   * playerID: INTEGER (Foreign key referencing Players table)
   * card1: VARCHAR(255)
   * card2: VARCHAR(255)
5. **BettingActions Table:** Primary Key: bettingActionsID Columns:
   * gameID: INTEGER (Foreign key referencing Games table)
   * roundNumber: INTEGER
   * playerID: INTEGER (Foreign key referencing Players table)
   * actionType: ENUM('bet', 'raise', 'fold', 'check')
   * actionAmount: DECIMAL(10,2)
6. **GameResults Table:** Primary Key: gameResultsID Columns:
   * gameID: INTEGER (Foreign key referencing Games table)
   * winningPlayerID: INTEGER (Foreign key referencing Players table)
   * winningHand: VARCHAR(255)
   * potAmountWon: DECIMAL(10,2)

**Relationships:**

1. One-to-Many: Players table to Games table (One player can participate in many games)
2. One-to-Many: Games table to CommunityCards table (One game has one set of community cards)
3. Many-to-Many: Games table to Players table (Many players can participate in one game)
4. Many-to-Many: Games table to BettingActions table (Many betting actions can occur in one game)
5. One-to-One: Games table to GameResults table (One game has one game result)

**Additional Considerations:**

1. Implement appropriate data types for each column to ensure data integrity.
2. Implement appropriate indexes for frequently queried columns to improve performance.
3. Consider implementing triggers to maintain data consistency across tables.
4. Implement appropriate security measures to protect sensitive data.

This database design should provide a solid foundation for managing data for a poker game application. It adheres to 3NF principles, ensuring data integrity and minimizing redundancy. The design can be further extended to accommodate additional features and requirements as needed.