# Database structure

cards

|  |  |  |
| --- | --- | --- |
| **field name** | **remarks** | **example value** |
| num | unique id of this pair of words | 1 |
| a | first word in pair | shoes |
| b | second word in pair | socks |

rooms

|  |  |  |
| --- | --- | --- |
| **field name** | **remarks** | **example value/possible values** |
| roomId | unique identifier |  |
| totalCount | total number of users in room |  |
| antiCount | number of undercover |  |
| blankCount | number of blank |  |
| hasStarted | has game started? |  |
| firstTurn | user index. first turn of current round. will be updated every round. | 0 |
| currentTurn | user index or string | 0/voting/hostVoting/ended |
| currentCount | number of users still alive |  |
| votes | array of user index who were voted out. order is not maintained. | [0,1,2,2] |
| usersWithMostVotes | array of user index who had the most votes (tie) | [0,1] |
| winner | user index or string.  if is user index, means only 1 single person wins then that person must be a blank. | 0/norm/anti/blank |
| users | array of user objects (refer to next table) |  |

rooms -> users

|  |  |  |
| --- | --- | --- |
| **field name** | **remarks** | **example value/possible values** |
| name | unique identifier |  |
| role | role in game | norm/anti/blank |
| card | given word | shoes |
| isHost | is host of game? |  |
| isOut | has been voted out? |  |
| hasVoted | has voted for current round? |  |

# How are users arranged in rooms.users array?

* every time a game is started, the rooms.users array will be shuffled and displayed in that order in the UI.

# When to end game and who wins?

* ~~when alive users > 2 and all alive users have same role, that role wins~~
* when all alive are norm, norm wins
* when alive anti > alive norm, antis win
* when alive users = 2
  + if there is blank alive, that blank wins
  + if there is anti alive, antis win