**ROOM**

Id: DataTypes.UUIDV4 //primary key

name: DataTypes.STRING, //name of the room  
maxPlayers: DataTypes.INTEGER,//max number of players   
createdAt: DataTypes.DATE,  
updatedAt: DataTypes.DATE

**USER**

id: DataTypes.UUIDV4 //primary key

roomId: DataTypes.UUIDV4, //foreign key - room the user is in

fbId: DataTypes.STRING, //facebook id (unique)

fbToken: DataTypes.STRING, //facebook auth token (unique)

name: DataTypes.STRING, //username

pic: DataTypes.STRING, // profile pic url

createdAt: DataTypes.DATE,

updatedAt: DataTypes.DATE

**CARD**

id: DataTypes.UUIDV4, //primary key

userId: DataTypes.UUIDV4, //creator of card

type: DataTypes.STRING, “white” or “black” or possibly others if we want

text: DataTypes.STRING, //text shown on the card

createdAt: DataTypes.DATE,

updatedAt: DataTypes.DATE

**VOTE**

id: DataTypes.UUIDV4 //primary key

(composite index- userId, cardId)

userId: DataTypes.UUIDV4 //creator of vote

cardId: DataTypes.UUIDV4 //card being voted on

upvoted: DataTypes.BOOLEAN, //true if user upvoted, false if user downvoted

createdAt: DataTypes.DATE,

updatedAt: DataTypes.DATE

**HOST**

id: DataTypes.UUIDV4 //primary key

(composite index - userId, roomId)

userId: DataTypes.UUIDV4 //user id of host

roomId: DataTypes.UUIDV4 //the host’s room id

createdAt: DataTypes.DATE,

updatedAt: DataTypes.DATE

**JUDGE**

id: DataTypes.UUIDV4 //primary key

(composite index - userId, roomId, place)

userId: DataTypes.UUIDV4 //user id of judge

roomId: DataTypes.UUIDV4 //the judge’s room id

place: DataTypes.INTEGER //place in rotation

createdAt: DataTypes.DATE,

updatedAt: DataTypes.DATE

**GAME**

id: DataTypes.UUIDV4 //primary key

roomId: DataTypes.UUIDV4 //room that is having the game

finishTime: DataTypes.DATE //time the game completed; null if still in pogress

createdAt: DataTypes.DATE

updatedAt: DataTypes.DATE

\*Note, to get current round of game, just get Rounds with the associated game id and pick the most recently created (sort by createdAt).

**HAND** - a players hand of cards in a game

id: type: DataTypes.UUID, //primary key

(composite index - cardId, gameId, userId)

cardId: DataTypes.UUID, //Card id of card in a hand

gameId: type: DataTypes.UUID, //Game id of game associate with the hand

userId: DataTypes.UUID, //User id of the user that has this hand

played: DataTypes.BOOLEAN, //true if user has played this card and is not still in the user’s

//hand, false if user has not played the card

createdAt: DataTypes.DATE,

updatedAt: DataTypes.DATE

**ROUND** - a round of a game

id: DataTypes.UUID, //primary key

gameId: DataTypes.UUID, //Game id of the game having this round

judge: DataTypes.UUID, //User id of the user that is the judge for this round

blackCard: DataTypes.UUID, //Card id of the black card that is used this round

winner: type: DataTypes.UUID, //User id of the winner of the round; null-round not over

winningCard: DataTypes.UUID, //Card id of the winning card of the round; null-round not over

state: DataTypes.STRING, //state of the round “waiting for players”, “waiting for judge”, “over”

createdAt: DataTypes.DATE,

updatedAt: DataTypes.DATE

**PLAYER STATE** - a players state during a game

id: DataTypes.UUIDV4, //primary key

userId: DataTypes.UUID, //User id of the user known as the player

gameId: DataTypes.UUID, //Game id of the game the player is

state: DataTypes.STRING, //state of player“pre-game”, “playing”, “waiting for players”,

//“waiting for judge”, “judging”, “round review”, “game review”

points: DataTypes.INTEGER, //how many round wins this game

place: DataTypes.INTEGER, //players place in the game

createdAt: DataTypes.DATE,

updatedAt: DataTypes.DATE

**PLAYED CARD** - a carded played during a round

id: type: DataTypes.UUID, //primary key

(composite index - userId, cardId, roundId)

userId: type: DataTypes.UUID, //User id of the user that played the card

cardId: DataTypes.UUID, //Card id of the card played

roundId: DataTypes.UUID, //Round id of the round the card was played

createdAt: DataTypes.DATE,

updatedAt: DataTypes.DATE