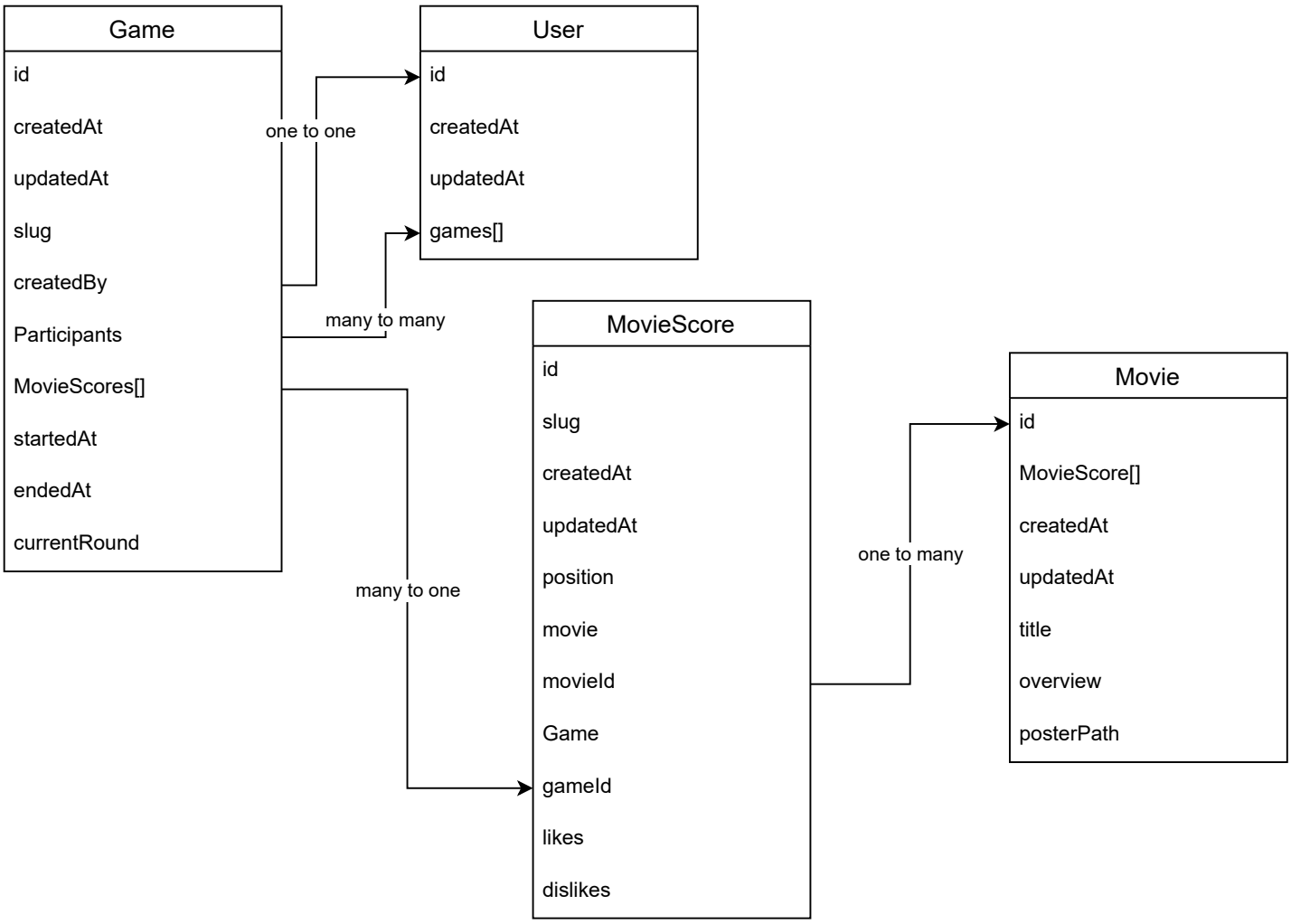

Data / Application Flow

1.) On every request the application should check for a session cookie.

FAILURE:

a.) create a new user in db
b.) create a session cookie for that user
c.) return userId to loader / action function
d.) continue

SUCCESS:

a.) return userId to loader / action function

b.) continue

2.) User Creates a Game (/games/create):

a.) get userId from cookie (1)

b.) fetch movies for game / sync in db

c.) create a new game in db, connect user as createdBy and Users[]

i.) data = gameStuff

ii.) createdBy: { connect: { user: userId } }

iii.) participants: [{connect: { user: userId } }]

iv.) movieScores: movies.map((movie, i) => ({ data: { likes: 0, dislikes: 0, position: i }, movie: { connect: { movie: movie.id } } }))

d.) return redirect to /games/:slug/lobby

3.) User Joins a Game (/games/:slug/lobby) (user arrives on page and clicks "join"

and logic is handled in action or users can autojoin using the loader

a.) get userId from cooke (1)

b.) get gameSlug from route params

c.) get game using slug, also select participants relation

d.) if user is not a participant in game connect user to game:

i.) update: { participants: [{ connect: { user: userId }] }

e.) return updated game and users list/count and maybe isCreator boolean

NOTE: there will need to be a code path in the loader to return a redirect to `/game/:slug/<active-game-route>` or

there will need to be redirect logic on the client side to navigate to a new route if `startedAt` timestamp is populated

4.) Waiting in lobby for game to start, only client side, should poll server every 1 second

or so if you want to show how many players have joined

5.) Creator Begins Game by pressing "Start" button in lobby

You need to know how long the round will last!!!

a.) get userId from cookie

b.) get game slug from route params

c.) update game with startedAt = Date.now()

d.) return updated game (if client navigating) or return redirect to `/game/:slug/round1

6.) User plays round 1 `/game/:gameSlug/:scoreSlug`

LOADER:

a.) get userId from cookie

b.) get score slug from route params

c.) select movieScore with scoreSlug with movie relation

d.) return list of movies for round 1 with needed data, movieScoreId, movie poster path, movie name

ACTION:

a.) get userId from cookie

b.) get movieScoreId from formData

d.) get like / dislike / actionType from formData

e.) get movieScoreId from formData

e.) update that movieScore by incrementing either totalLikes or totalDislikes

7.) Creator reaches the end of round

ACTION:

a.) get userID from cookie

b.) get game slug from route params

c.) update game and increment round

d.) redirect to `/game/:slug/round2` if moving directly into next round

OR redirect to /game/:slug/round1/results to show a results of the round and there you could start round 2 via user action