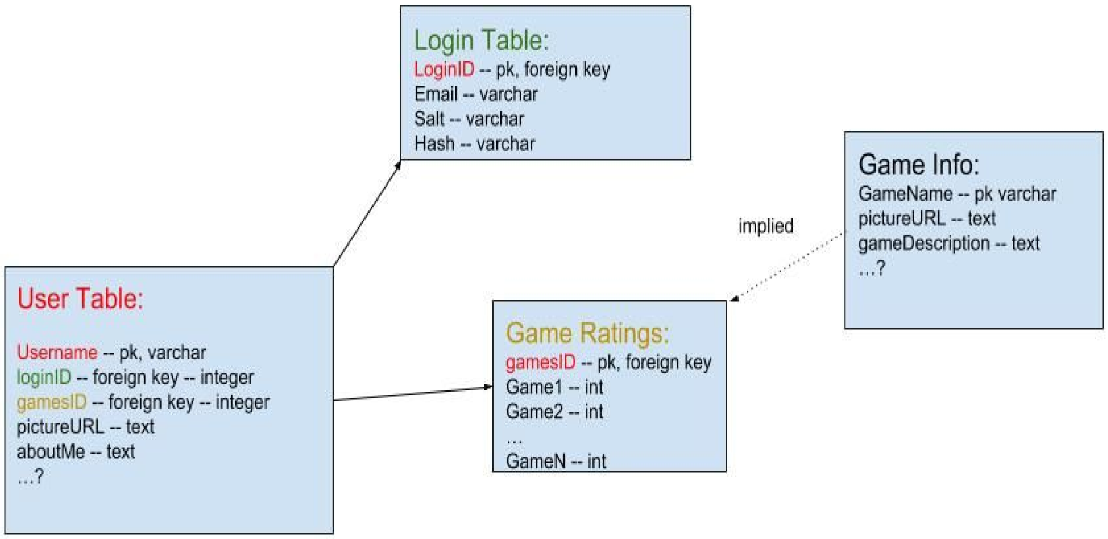
Milestone 4
--draft

Table diagram (large one included in folder)



Logic behind tables:

Since we are making a game recommendation website, the schema needs to be organized in a way where the data needed for generating user recommendations is as accessible as possible. To recommend users games we essentially just need a matrix of game ratings where each row (in this case) corresponds to a specific user and each column corresponds to a specific game. So the game ratings table is organized in that manner.

The user table is basically the access point to all other tables. From the user table we can look up the games the user has rated and we can also find the user's login information. The table also contains other data that will be relevant on the user's page (such as a profile picture or a user written description of themselves).

The login table is connected to the user table via the login ID. While the information could have perhaps been included in the user table it is preferable to have it in a separate table. We wanted the schema to follow the principle of separation of concerns, so that each table exists for a specific function. The user table is a hub, the game table is for the recommendation algorithm, etc. having the user table handle both login and navigation would violate this principle.

The game info table is a standalone table, but implicitly it is linked to the ratings table. There doesn't seem to be a good way to have the name of a column work as a key, so the implicit linking will be handled in the middle layer.

Management software. We will be using postgres sql.

note:
At the moment, the game names will just be letter names. This should allow us to conduct necessary program tests. I'm still having trouble finding game data that is a good size and in a useable format. I will be looking deeper into a few APIs that seem promising over fall break, if those don't seem to work I will need to build a web scraper or enter data manually.