Web Application Entity Design

Checkers 667

Team D

Milestone 3

Team Members:

1. Liszewski Ryan
2. Salgado Rodolfo
3. Singh Sukhjit
4. Niftiyev Hasan

GitHub Repository Link:

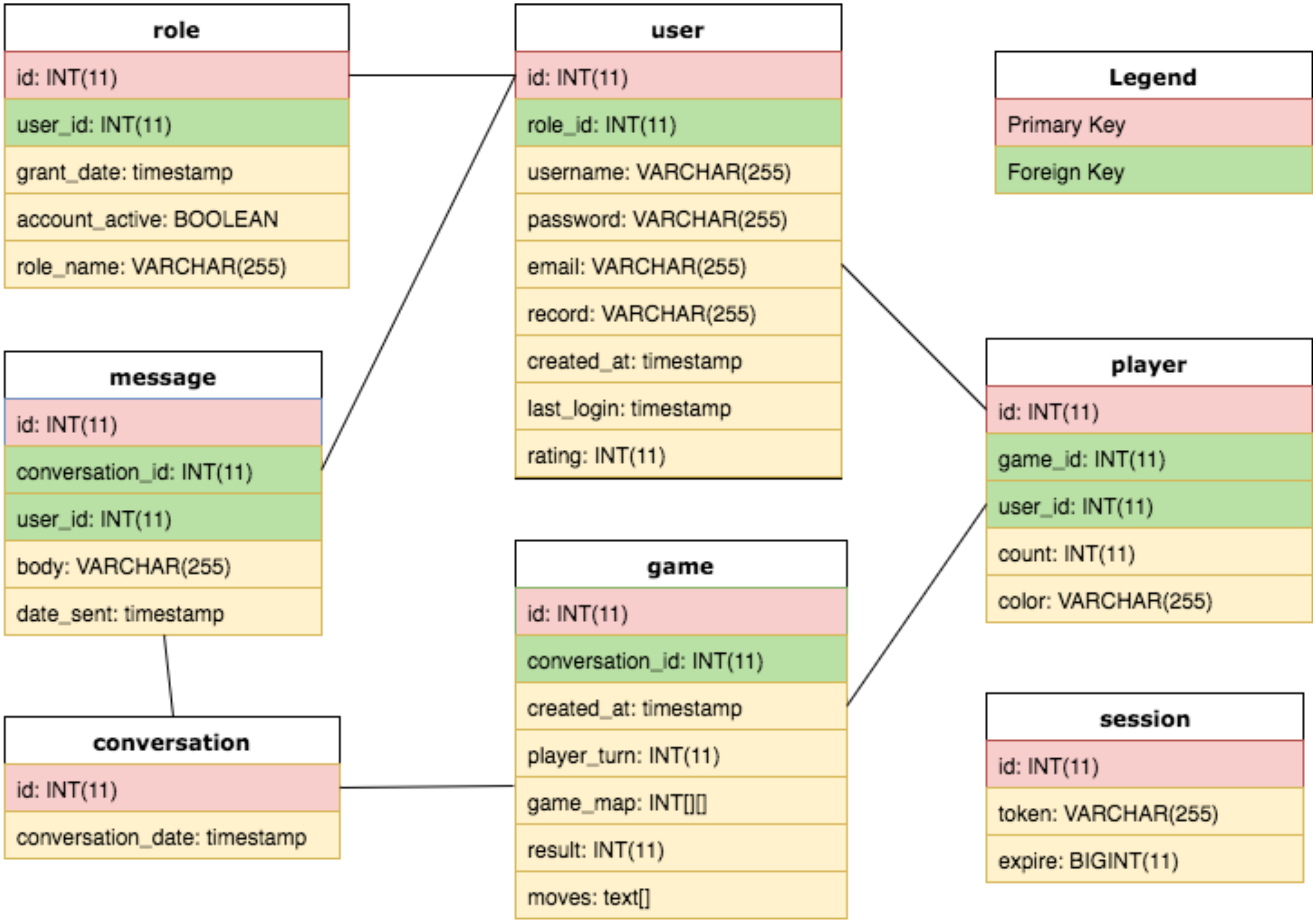
https://github.com/NiftiyevHasan/checkers667

Table of Contents

1. Entity Diagram

2. Entity Descriptions

1. Entity Diagram



role
id: INT(11)
user_id: INT(11)
grant_date: timestamp
account_active: BOOLEAN
role_name: VARCHAR(255)

user
id: INT(11)
role_id: INT(11)
username: VARCHAR(255)
password: VARCHAR(255)
email: VARCHAR(255)
record: VARCHAR(255)
created_at: timestamp
last_login: timestamp
rating: INT(11)

Legend
🟥 Primary Key
🟩 Foreign Key

message
id: INT(11)
conversation_id: INT(11)
user_id: INT(11)
body: VARCHAR(255)
date_sent: timestamp

player
id: INT(11)
game_id: INT(11)
user_id: INT(11)
count: INT(11)
color: VARCHAR(255)

game
id: INT(11)
conversation_id: INT(11)
created_at: timestamp
player_turn: INT(11)
game_map: INT[][]
result: INT(11)
moves: text[]

conversation
id: INT(11)
conversation_date: timestamp

session
id: INT(11)
token: VARCHAR(255)
expire: BIGINT(11)

2. Entity Descriptions

1. **Session**
    a. Identifies a single user session.
2. **Role**
    a. Grant’s a type of permission to an User. There will be three rolls, including:
        i. Admin
        ii. Spectator
        iii. Player
    b. Monitor’s if the user’s account is active or not.
3. **User**
    a. Contains all the data of a user, including their:
        i. Username
        ii. Password
        iii. Email
        iv. Their record (Win - Loss)
        v. Their rating
        vi. A timestamp when the user last logged in and when the user registered.
    b. A user can then access their data, update it such as changing their username, and also view other user’s information (read-only).
4. **Player**
    a. Represents a single user in a game.

b. Holds the count of the number of pieces a player has remaining. Once the player's pieces count goes to zero, then that player loses the game.

c. Knows which color the player is (Black or red).

   i. Tells which player goes first in the game.

   ii. Tells the game the the positioning of each player of the game

      1. Top and Bottom of the board.

5. Game

a. Keeps track of the checker game by using a 2D integer array to represent the checker's board and each player's pieces.

b. Which player's turn it is.

c. The result of the game.

   i. Which player won.

d. All the moves of the game in a string array. Each move consists of 15 characters, were the first four characters are the start point of the move, and the end point of the move, respectively. The last 11 characters are is the player's id who made the move.

   i. For example, a move by player "1" from E4 to F5 would be "E4F500000000001"

e. The conversation of the game by having a conversation foreign key.

6. Message

a. Consists of a single message that consists of:

   i. The time at which the message was sent.

ii. The user who sent the message.

iii. The text of the message.

b. The message belongs to a conversation by having a conversation’s foreign key.

7. Conversation

a. Embodies conversation that is composed of a group of messages.

b. A single conversation will be used for the lobby chat between users that will be defined on the client side.

c. A new conversation will be created when a new game is created.