Maxim Gura 222062 https://gitlab.com/Maks_Gura/bds-db-design.git

Pavel Melnik 221558 https://gitlab.com/PavelMelnik/bds_db_design.git

For our project, we decided to implement the database of the football matches website according to the rules of the league.
Each registered user can make predictions about events in a certain match. For example, number of passes, kicks per goal, removal of players from field, number of goals scored per team, total number of goals scored, number of assists in a match, etc. These predictions are compared with actual results of the match and users get points.
New users have to enter to the database the following information: name, surname, address, e-mail and a unique nickname that will be used on the website.

In the match, two teams play against each other according to the rules of a certain football league. The website (and database) contains information about them: team name, number of players in the team, team coach, team manager, each player's manager and coach's manager.

Also, in the database there is an information about the stadium of the team (where the match will be played): the address of the stadium and which team it belongs to.

The database contains next information about managers (team manager, coach manager and player manager): name of the manager, surname and e-mail.

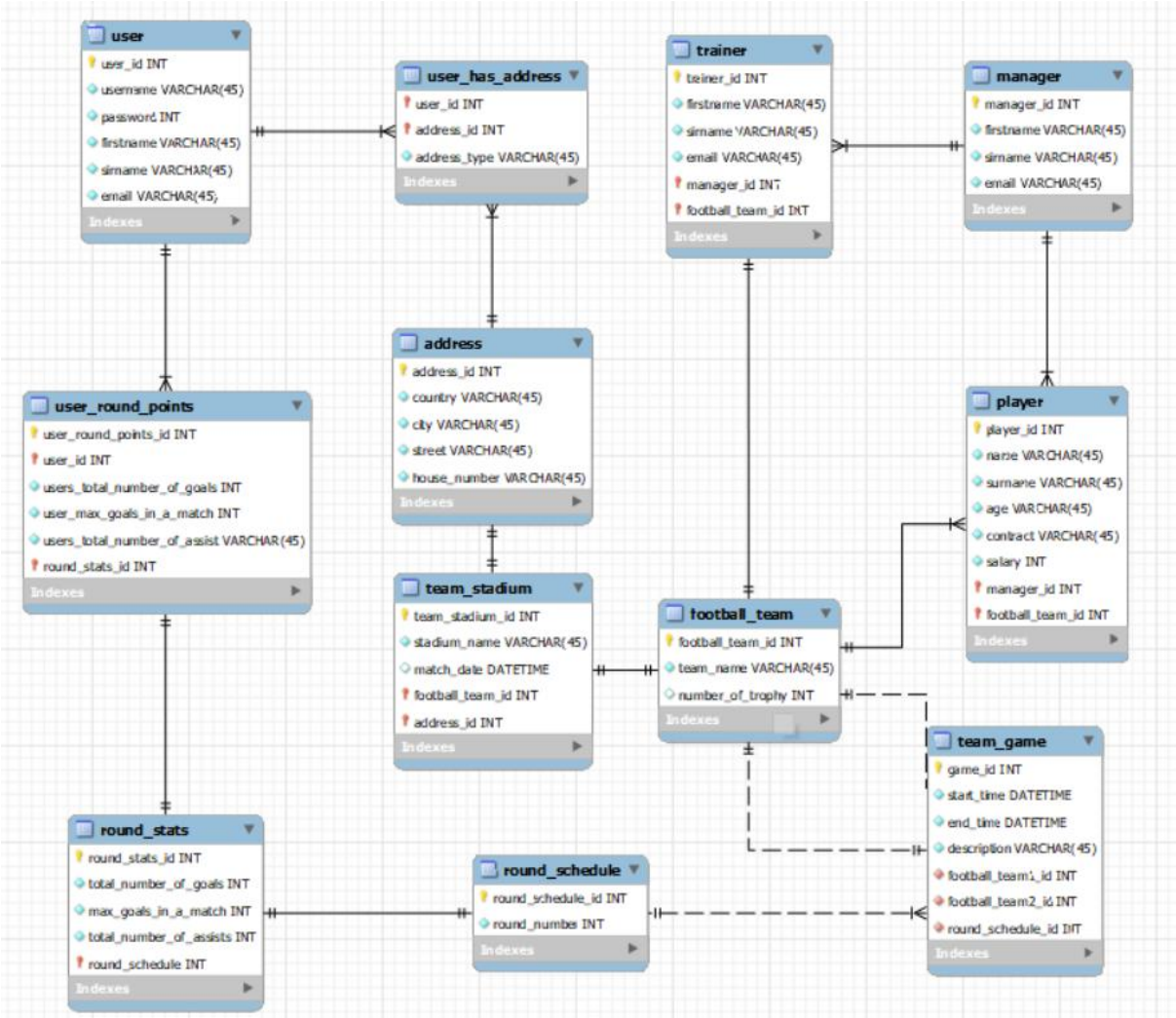
The player table consists of player name, surname, age, date of completion of the contract and the salary of the player.

The table of the game (team_table) includes: date of starting and ending of the game, description of the game (which teams will play)) and the schedule of the game.

The trainer table has the information about the name and surname of the trainer, his e-mail, manager ID, trainer ID as a primary key and football team ID as a secondary key.

Points that user can win if his prediction is successful, will be put in the “user_round_points” table. This table consists of user_round_points_id as a primary key, round_stats_id as a secondary key, total number of goals scored, number of goals in the game, and number of assists in the game.

The “round_stats” table is responsible for representation of real (not predicted) results of the game in the database. This table includes: “round_stats_id” as a primary key, “round_schedule_id” as a secondary key, number of goals scored, number of assists.


user
user_id INT
username VARCHAR(45)
password INT
firstname VARCHAR(45)
surname VARCHAR(45)
email VARCHAR(45)
Indexes
user_has_address
user_id INT
address_id INT
address_type VARCHAR(45)
Indexes
trainer
teiner_id INT
firstname VARCHAR(45)
surname VARCHAR(45)
email VARCHAR(45)
manager_id INT
football_team_id INT
Indexes
manager
manager_id INT
firstname VARCHAR(45)
surname VARCHAR(45)
email VARCHAR(45)
Indexes
address
address_id INT
country VARCHAR(45)
city VARCHAR(45)
street VARCHAR(45)
house_number VARCHAR(45)
Indexes
user_round_points
user_round_points_id INT
user_id INT
users_total_number_of_goals INT
user_max_goals_in_a_match INT
users_total_number_of_assist VARCHAR(45)
round_stats_id INT
Indexes
player
player_id INT
name VARCHAR(45)
surname VARCHAR(45)
age VARCHAR(45)
contract VARCHAR(45)
salary INT
manager_id INT
football_team_id INT
Indexes
team_stadium
team_stadium_id INT
stadium_name VARCHAR(45)
match_date DATETIME
football_team_id INT
address_id INT
Indexes
football_team
football_team_id INT
team_name VARCHAR(45)
number_of_trophy INT
Indexes
team_game
game_id INT
start_time DATETIME
end_time DATETIME
description VARCHAR(45)
football_team1_id INT
football_team2_id INT
round_schedule_id INT
Indexes
round_stats
round_stats_id INT
total_number_of_goals INT
max_goals_in_a_match INT
total_number_of_assists INT
round_schedule INT
Indexes
round_schedule
round_schedule_id INT
round_numbe INT
Indexes