

# Modelo Conceitual

## Introdução

O objetivo é criar um banco de dados que armazene o resultado de partidas do jogo Dota 2. Dota significa Defesa do Ancestral (*Defence of The Ancient*), onde vence aquele que derrubar as estruturas do inimigo primeiro. Este jogo é um MOBA (*Multiplayer Online Battle Arena* – Arena de Batalha Online de Multijogadores), o qual 2 times se enfrentam, com 5 jogadores para cada time. Os jogadores escolhem Heróis (personagens) que desejam jogar, cada herói tem de 5 a 6 habilidades. No período inicial do jogo, os jogadores se distribuem em 3 trilhas, chamadas de *Top*, *Bot* e *Middle*. Além disso, com o objetivo de obter vantagem no jogo e vencer, os jogadores compram itens para seus personagens com o ouro que obtém eliminando criaturas e outros jogadores. Ainda, é possível utilizar várias estratégias para vencer o jogo como a conquistas de objetivos dentro do jogo e ainda, destruir estruturas de forma rápida.

Os jogadores podem jogar partidas casuais, ou então, jogar partidas competitivas nos servidores das regiões, com objetivo de conquistar pontuação, denominada mmr. Esta pontuação indica a habilidade do jogador e também serve para classificar os jogadores, sendo que o rank 1 é o melhor jogador de uma região específica.

Como um grande jogo online, Dota 2 possui sistema competitivo profissional. Este banco de dados irá coletar dados dos jogadores de times profissionais específicos. Uma vez obtido o ID dos jogadores, serão coletados informações das partidas dos jogadores, com o objetivo principal de se ter estatística dos Heróis em relação as lanes e seu percentual de vitória, além de outras observações que serão feitas ao longo deste projeto.

Neste banco não será incluído os itens de cada herói por partida, tão pouco itens cosméticos utilizados.

Os dados serão obtidos através da [OpenDota API \(20.0.0\)](#), filtrados e tratados em python e então adicionados ao banco de dados. As consultas serão feitas utilizando o MySQL Workbench.

A baixo segue a o diagrama entidade-relacionamento representado pela figura 1.

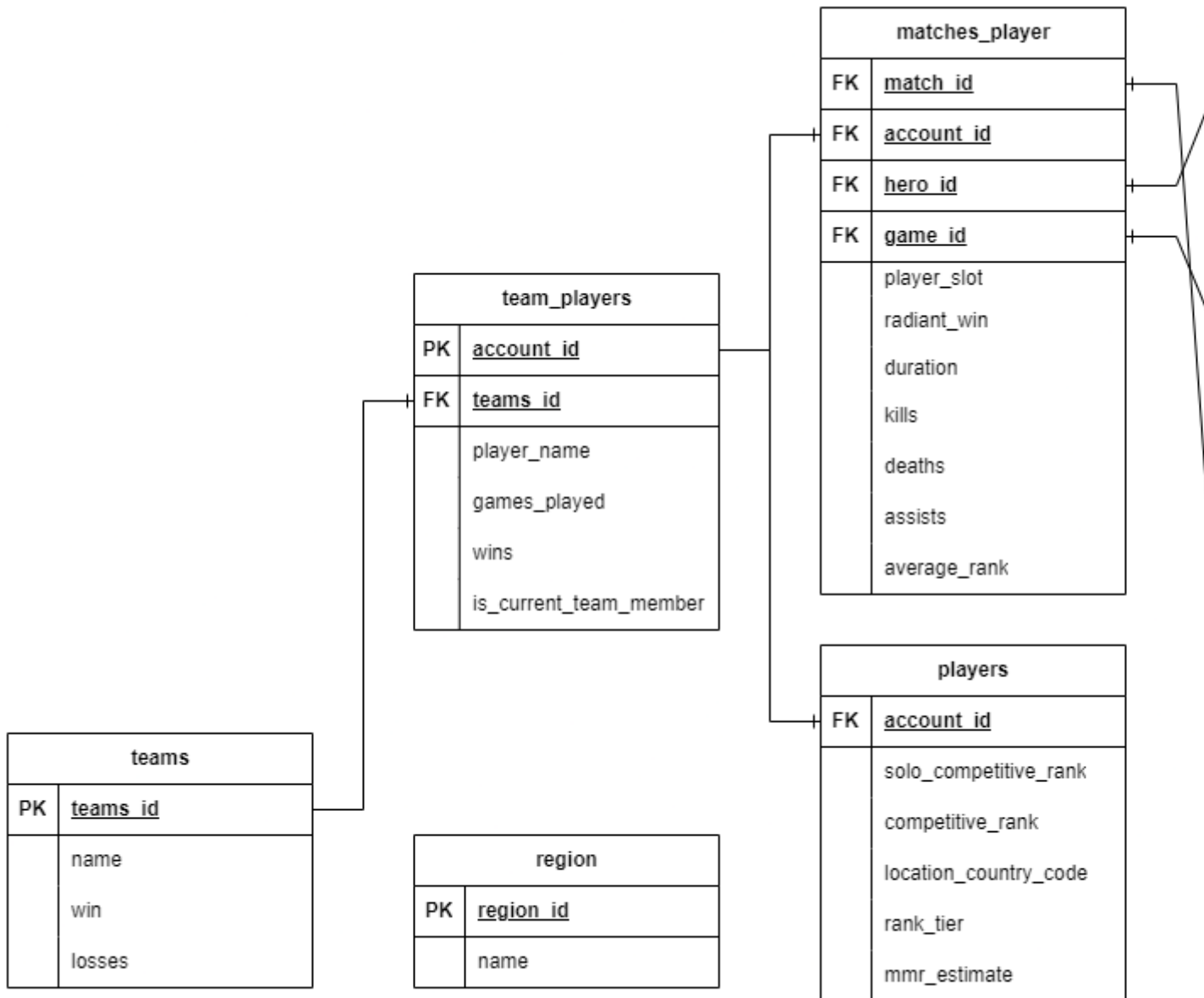


Figura 1 – Diagrama entidade-relacionamento.

## Modelo Lógico

O objetivo neste capítulo é definir as características das colunas existentes nas tabelas do database, bem como, suas restrições.

### Tabelas

heroes:

- id (int) – Chave Primária – Valor Único – Não Nulo;
- name (varchar) – Valor Único;

region:

- id (int) – Chave Primária - Valor Único – Não Nulo;
- name (varchar) – Valor Único – Não Nulo;

game\_mode:

- id (int) – Chave Primária – Valor Único – Não Nulo;
- name (varchar) – Valor Único – Não Nulo;
- balanced (boolean);

teams:

- id (int) – Chave Primária – Valor Único – Não Nulo;
- name (varchar) – Valor Único – Não nulo;
- win (int);
- losses (int).

team\_players

- account\_id (int) – Chave Primária – Valor Único – Não Nulo;
- teams\_id (int) – Chave Estrangeira;
- player\_name (varchar) – Valor Único;

- games\_played (int);
- wins (int);
- current\_team\_member (Boolean).

players:

- account\_id (int) – Chave Estrangeira – Valor único;
- solo\_competitive\_rank (int);
- competitive\_rank (int);
- location\_country\_code (int) – Chave Estrangeira;
- rank\_tier (int);
- mmr\_estimate (int).

matches

- id (int) – Chave Primária – Not Null;
- game\_id (int) – Chave Estrangeira;
- barracks\_status\_dire (int);
- barracks\_status\_radiant (int);
- dire\_score (int);
- radiant\_score (int);
- radiant\_gold\_adv (int);
- tower\_status\_dire (int);
- tower\_status\_radiant (int);

matches\_players:

- match\_id (int) – Chave Estrangeira – Não Nulo;
- account\_id (int) – Chave Estrangeira - Não Nulo;
- hero\_id (int) – Chave Estrangeira – Não Nulo;

- game\_id (int) – Chave Estrangeira;
- player\_slot (int);
- radiant\_win (bool);
- duration (int);
- kills (int);
- deaths (int);
- assists (int);
- average\_rank (decimal).

## **Modelo Físico**

O schema dota2 foi criado no MySQL Workbench e os comandos utilizados estão descritos no documento CreateDataBase.sql.

Os dados foram adquiridos via api do dota, no endereço eletrônico <https://docs.opendota.com/>, utilizando python. As funções utilizadas para fazer o request na api estão na pasta apiDota no arquivo apiConsume.py. As funções utilizadas para tratamento e inserção dos dados no schema dota2, estão descritas no arquivo schemaConnection.py. Para fins de melhor aproveitamento de memória RAM e debug do código para inserção das partidas na tabela Matches, foi criado um arquivo separado de inserção chamado de InsertMatches, que utiliza dados da planilha matches.csv.

O notebook dataManipulation.ipynb é o local onde as funções acima descritas foram utilizadas para adquirir os dados e então inserir no schema dota2.

O diagrama gerado pelo MySQL Workbench está salvo no documento eerdota2.mwb.

Quanto aos softwares passíveis de uso para este projeto, PostgreSQL por conta do bom suporte para consultas e pela sua capacidade em suportar grandes volumes e cargas de dados, já que esse possível um dia, pode ser escalado. Quanto ao Oracle e ao SQL Server não seriam utilizados pela necessidade de contratação de licença.