Olympics Athletes vs. Pokémon: Project Milestone 2

Member: Chuyang Deng, Luo Na, and Yan Zhong
Github link: github.com/ChuyangDeng/athletesvspokemon

Project description and motivation:

Because of the exciting Olympics Game in the summer of 2016 and the recent popularity of *Pokémon GO*, our team decided to design and develop a web application that can match Olympics athletes with Pokémon. For each athlete, we will find a best matched Pokémon based on his/her height, weight, sports, and number of medals earned. The detailed algorithm is described below. Besides entertainment, this website also provides kids an opportunity to learn about the Olympics in a fun way through a game they enjoy.

Datasets

We will be using the following datasets: (1) Olympic medal winners: every one since 1896 (to 2008), (2) Olympic data for 2012 and 2016, (3) Athletes' information for recent years, (4) Nintendo Pokedex for all pocket monsters from six generations, (5) pictures and descriptions of athletes and Pokémon in noSQL component.

Dataset #1 only includes information about medal winners from 1896 to 2008, so the recent two years' medal winners' information will be supplemented by #2. #2 and #3 will provide athletes' body weight and height that #1 does not include. #4 provides sufficient information about Pokémon. We will also use noSQL solution to provide pictures and detailed descriptions of athletes and Pokémon.

Matching Algorithm

First, we will match the sports in Olympics with Pokémon's categories, as shown in Table 1. These matches are certainly not perfect, but each pair does share some common features (e.g. in water, in air). Then we will use the athlete's body weight and height to match with Pokémon's HP value. (W + H and HP will be properly scaled and directly match in query.) Finally, based on the number of Olympic medals earned by athletes (with Gold worth more points), we can generate a rank for all athletes (in each category). And based on the overall score of Pokémon (provided in dataset #4), we can generate a rank for Pokémon as well. By matching these two lists of ranking, we can find a Pokémon that can properly represent a certain athlete. The result will be displayed with pictures and other detailed description (potentially from Wikipedia).

Features

The above matching features will definitely be implemented in this application. Currently we plan to only include several recent Olympic games instead of every game since 1896. But we will analyze more games if we have enough time.

Also, we can potentially include winter Olympics in our application to increase the diversity of sports and Pokémon categories. We might also find out the most popular Pokémon and the most popular athletes from twitter feeds.

Technologies:

- AWS to host datasets and website
- Node.js (Environment to run Express, back end)
- Express (Set up server)
- AngularJS (Implement User Interface, front end)
- Basically MEAN stack without MongoDB, but we will use noSQL.

Preliminary division of responsibilities:
Na Luo: AngularJS, Front end
Chuyang Deng: Database design/population
Yan Zhong: Node.js/backend/server

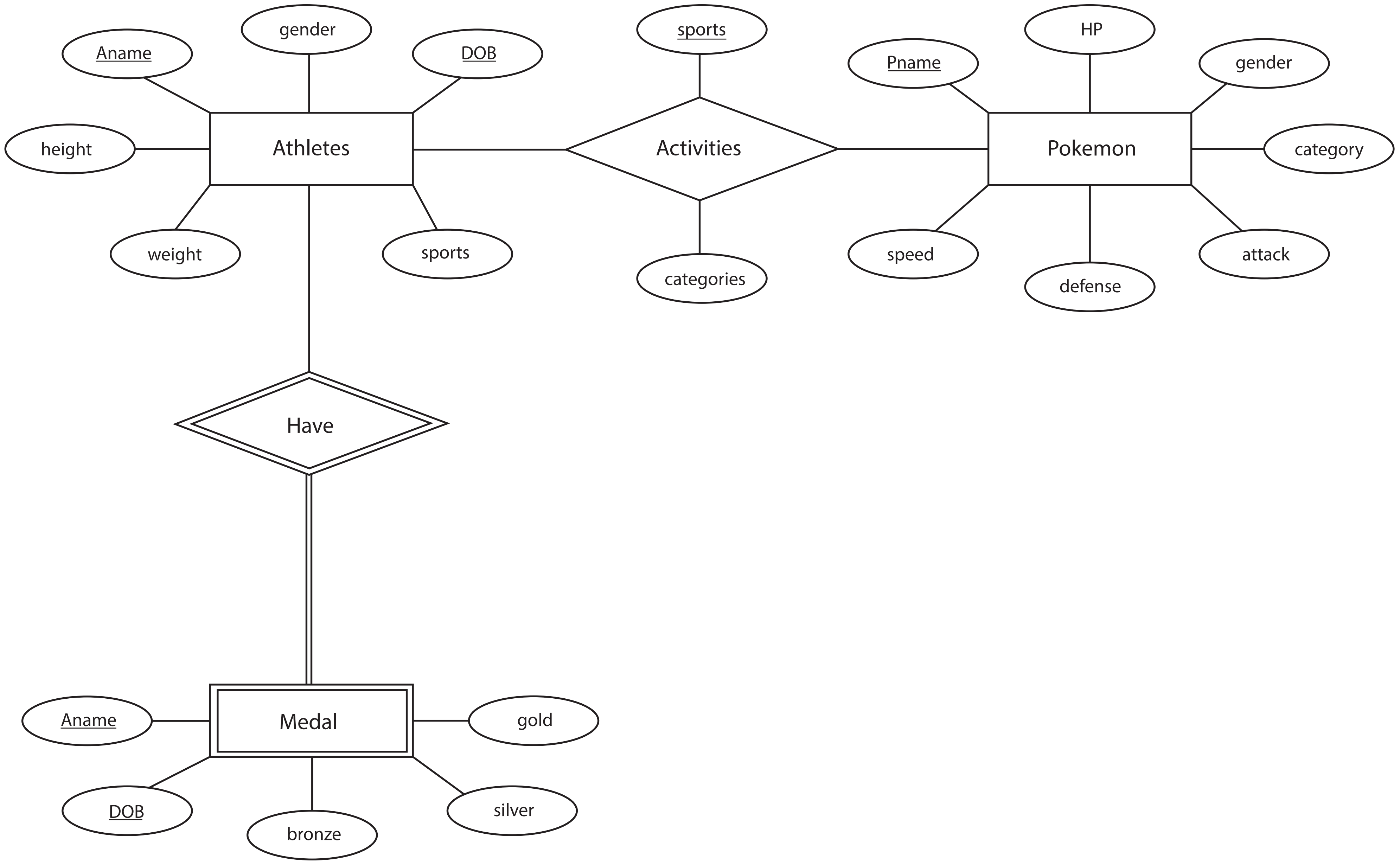
Schema

Athletes (name, DOB, height, weight, gender, sports)
Medal (name, DOB, gold, silver, bronze) <weak entity of Athletes>
Activities (Sports, Categories)
Pokemon (name, HP, total, speed, defense, attack, gender, category)

SQL DDL

```
-- Athletes Relation
CREATE TABLE Athletes {
        name VARCHAR(255) NOT NULL,
        DOB VARCHAR(255) NOT NULL,
        height NUMBER,
        weight NUMBER,
        gender VARCHAR(255),
        sports VARCHAR(255),

        PRIMARY KEY (name, DOB),
        CHECK gender IN ('male', 'female')
}

-- Medal Relation (weak entity of Athletes)
CREATE TABLE Medal {
        name VARCHAR(255) NOT NULL,
        DOB VARCHAR(255) NOT NULL,
        gold INTEGER,
        silver INTEGER,
        bronze INTEGER,

        PRIMARY KEY (name, DOB),
        FOREIGN KEY name REFERENCES
Athletes(name),
        FOREIGN KEY DOB REFERENCES
Athletes(DOB),
        ON DELETE CASCADE
}
```

```
-- Pokemon Relation
CREATE TABLE Pokemon {
        name VARCHAR(255) NOT NULL,
        HP INTEGER,
        total INTEGER,
        speed INTEGER,
        defense INTEGER,
        attack INTEGER,
        gender VARCHAR(255),
        category VARCHAR(255),

        PRIMARY KEY (name)
}

-- Activites Relation
CREATE TABLE Activites {
        sport VARCHAR(255) NOT NULL,
        category VARCHAR(255),

        PRIMARY KEY (sport),
        FOREIGN KEY sport REFERENCES
Athletes(sport),
        FOREIGN KEY category
REFERENCES Pokemon(category)
}
```


Aname
gender
DOB
height
Athletes
weight
sports
sports
Activities
categories
Pname
HP
gender
Pokemon
category
speed
defense
attack
Have
Aname
Medal
gold
DOB
bronze
silver