CIS 550 - Database and Information Systems

Project Milestone 2

Olympikedia: Web Application

Team Members - Project Group 14

- Abhinav Malik
- Shivendra Pandey
- Devanshu Jain
- Sarath Vadakkepat

Motivation

In the 2016 summer Olympics a total of over 11,000 athletes from 207 National Olympic Committees took part with 306 set of medals, featuring 28 Olympic Sports. They were spread over 33 venues in the host city and gathering information for all these event is an arduous task.

To make it easy we decided to implement a wikipedia for Olympics, titled Olympikedia, which aims at providing all the information one could possibly need, either for the Summer Olympics or for the Winter Olympics. Using Olympikedia we'll be able answer all the possible queries for the current Olympics or any Olympics in the history of the event.

Features

Features of this project will include:

- **Search Form**
  The search form will enable users to get all information by selecting particular editions of the game. The user can optionally select particular sporting events that were organised in that game. The user can:
  1) View the final medal tally of that edition of Olympic Games
  2) View all the sporting events that were organised in that game (if that information was not entered while submitting the search form)
  3) View the results of each sporting event - The result will include the country and the athlete(s) depending on whether it was a team sport or individual.
  4) View a particular athlete's information:
     a) Personal Information
     b) Pictures
     c) Medal Tally in all editions of Olympic Games
     d) Year-wise performance
  5) View a particular country's information:
     a) Medal Tally in all editions of Olympic Games

b) Year-wise performance

Sometimes, the user is more interested in looking at a particular country's performance instead of particular edition of the games. Although this can be achieved by going to a particular olympic event in which that country has won a medal and clicking on that country but we would like to reduce the number of clicks by adding a search form where the user can search by the country.

- **Medal Table**
  Overview of the medal table which can either be displayed country wise or sports wise.
- **Customized graphs**
  While showing the year-wise distribution of medals won by a country/individual, we will display it in a graphical format with a line graph.
  For Countries :
  1) Number of Gold/Silver/Bronze medals won as of current year vs previous years.

  For Participants :
  1) Medals won this year vs previous years
  2) Sport specific graphs (for eg: 100m running time this year vs past)
- **Live Medal Tally**
  We will also show a real time count of the ongoing Olympics games' medal tally split according to countries and the number of gold, silver and bronze medals using resources such as: here.
- **Notifications**
  If someone subscribes to daily updates then that person will get regular updates when and if their team or favorite athlete wins a medal. This can be done using Twilio API.

Tentative Features

- **Mobile phone application**
  To achieve it's maximum potential our application must be portable and should be user friendly. The best way to achieve this is a mobile application which provides a personal touch with an interactive user interface along with increased accessibility and portability.
- **Comparisons**
  We'll be using Winter Olympics dataset as the complementary dataset and hence will allow the user to compare between the two editions. We'll also be able to form our own custom comparisons and hence providing more power to the users.
- **Wikipedia**
  We will extract personal information about athletes from wikipedia articles and we will include those details in the athlete's page.

Technologies

- Web design- HTML, CSS, Javascript, jQuery, AJAX, Bootstrap
- MEAN stack
- Database Backend - MySQL, MongoDB
- Amazon Web Services
- Wolfram Alpha
- NodeJS

Member Responsibility

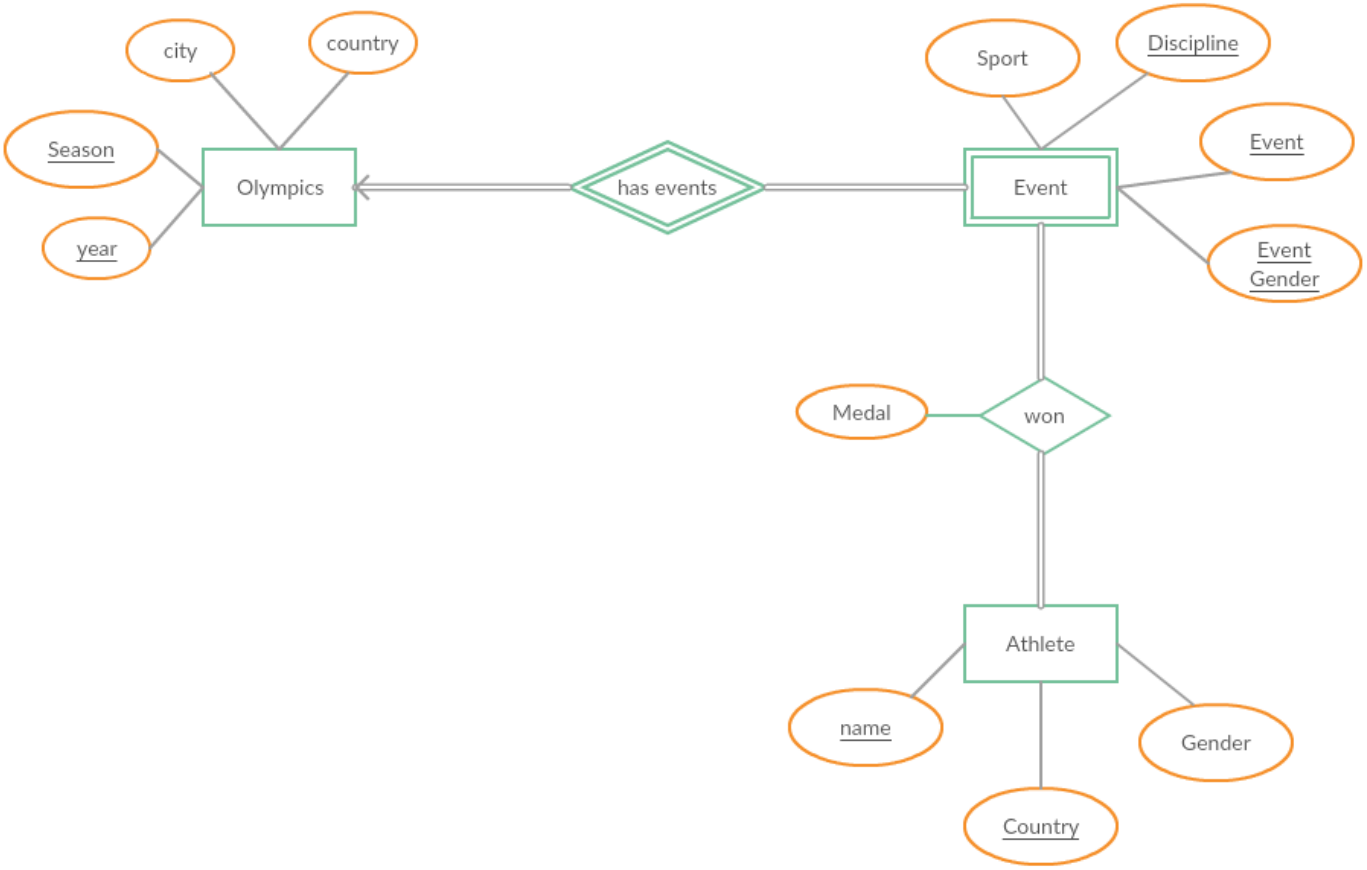
- Abhinav Malik
    1) Query Optimization
    2) Data Cleaning
- Shivendra Pandey
    1) Database Normalisation and Schema Design
    2) Graphical User Interface UI/UX Design
- Devanshu Jain
    1) Data Population
    2) Graphical User Interface UI/UX Design
- Sarath Vadakkepat
    1) Graphical User Interface UI/UX Design
    2) Data Population

Complementary Datasets

In the end, we will be referring to the following datasets:

- Winter Olympics dataset (here)
    1) To get the medal information for each edition of winter olympics
- Summer Olympics dataset (here)
    1) To get the medal information for each edition of summer olympics
    2) To get the athlete's information for each medal won
- Wikipedia
    1) To get athlete's personal information
- Database Olympics (here)
    1) To get the athlete's information for each medal won in winter olympics
- Olympics Official Website (here)
    1) To get the athlete's information like pictures, news articles, etc.

Entity Relationship Diagram



In addition, we are thinking of storing athlete's personal information in a NoSql database - MongoDB. Since, we may find different fields for different athletes and may even find none for some athletes, we don't have a fixed schema for this at this point of time.

Relational Schema

- **Olympics**(Year, Season, City, Country)
- **Event**(Year, Season, Event, Discipline, Event_Gender, Sport)
- **Athlete**(Name, Country, Gender)
- **Athlete_won**(Year, Season, Event, Discipline, Name, Country, Medal)

Note: We will only be storing those athletes who have won at least one medal in at least 1 event of any edition of olympic games.

SQL DDL Statements

```
CREATE TABLE Olympics (
        Year VARCHAR(10),
        Season VARCHAR(10),
        City VARCHAR(255) NOT NULL,
        Country VARCHAR(255) NOT NULL,
        CONSTRAINT chk_season CHECK (Season IN ('Summer', 'Winter')),
        PRIMARY KEY (Year, Season)
)

CREATE TABLE Events (
        Year VARCHAR(10),
        Season VARCHAR(10),
        Event VARCHAR(255),
        Discipline VARCHAR(255),
        Event_Gender VARCHAR(10),
        Sport VARCHAR(255) NOT NULL,
        CONSTRAINT chk_event_gender CHECK (LCASE(Event_Gender) IN ('m', 'w', 'x')),
        PRIMARY KEY (Year, Season, Event, Discipline, Event_Gender),
        FOREIGN KEY (Year, Season) REFERENCES Olympics ON DELETE CASCADE
)

CREATE TABLE Athlete (
        Name VARCHAR(50),
        Country VARCHAR(255),
        Gender VARCHAR(10) NOT NULL,
        CONSTRAINT chk_gender CHECK (LCASE(Gender) IN ('m', 'w')),
        PRIMARY KEY(Name, Country)
)

CREATE TABLE Athlete_won (
        Year VARCHAR(10),
        Season VARCHAR(10),
        Event VARCHAR(255),
        Event_Gender VARCHAR(10),
        Discipline VARCHAR(255),
        Name VARCHAR(50),
```

```
    Country VARCHAR(255),
    Medal VARCHAR(10) NOT NULL,
    CONSTRAINT chk_medal CHECK (LCASE(Medal) IN ('g','s','b')),
    PRIMARY KEY (Year, Season, Event, Event_Gender, Discipline, Name, Country),
    FOREIGN KEY (Year, Season, Event, Event_Gender, Discipline) REFERENCES Events
    ON DELETE CASCADE,
    FOREIGN KEY (Name, Country) REFERENCES Athlete ON DELETE CASCADE
)
```