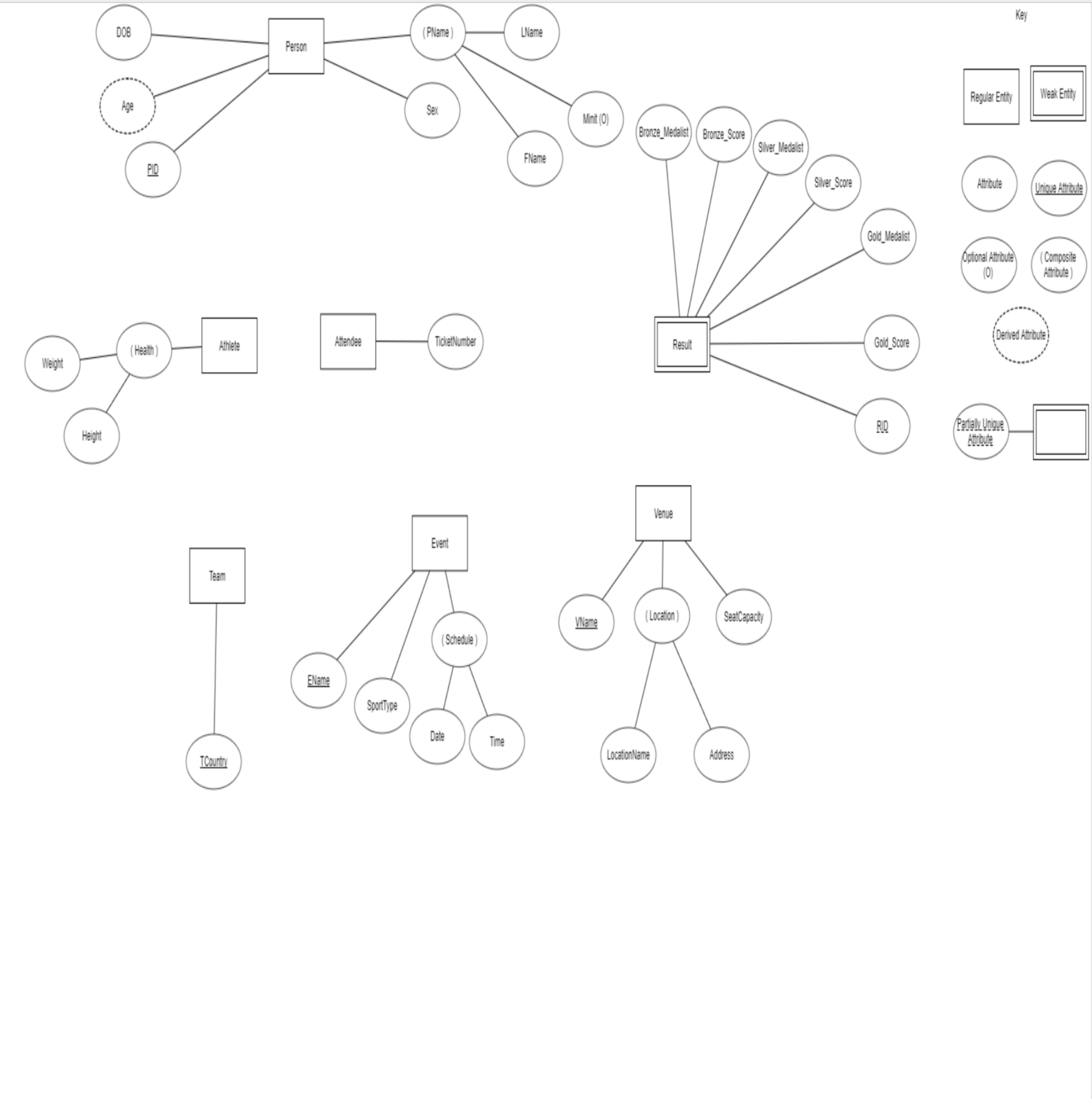
Overview

Our database stores and collects preliminary information for a promotional website for the Tokyo 2020 Summer Olympics, also known as the XXXII Olympic Games. The database stores information regarding the countries whose team will be participating in the games and personal information of their respective team members such as athletes and coaches. In addition, the database stores information regarding the sport type, name, date, and time of the events as well as information about the different venues where the events are located. Visitors to the site will be able to view and search the list of teams attending the games and read a brief description of the athletes. They will be also able to view and search a schedule of all the events and their respective specific location. Up-to-date information about event results such as scores and medals will also be available to view. Although anyone can view and search the database, only members of the International Olympic Committee (IOC) (admins to the site) can modify the database. The database should be used by IOC officials to insert new information as new qualifying teams and athletes are announced; they can also delete existing teams and/or team members who disqualify for whatever reasons that might occur. IOC officials can also use the database to plan and schedule for events. They can set a date, time, and specific venue for each event as well as keep count of the number of prospective attendees, such as general ticket holders, sponsors, and IOC officials, for each event. To add on, IOC officials can update information about the events to fit adjustments in the schedule as needed and can add the scores and medal recipients for each event once they are announced.

Single Entity Models



Relationships

1. **Athlete belongs to Team: Each athlete belongs to a team that represents their respective country at the Olympics. An athlete can only belong to one country while a country's team can represent one or more athletes. For example, Ryan Lochte, Simone Biles, and Cheryl Reeve are represented by Team USA. This a one-to-many (1:N) relationship.**

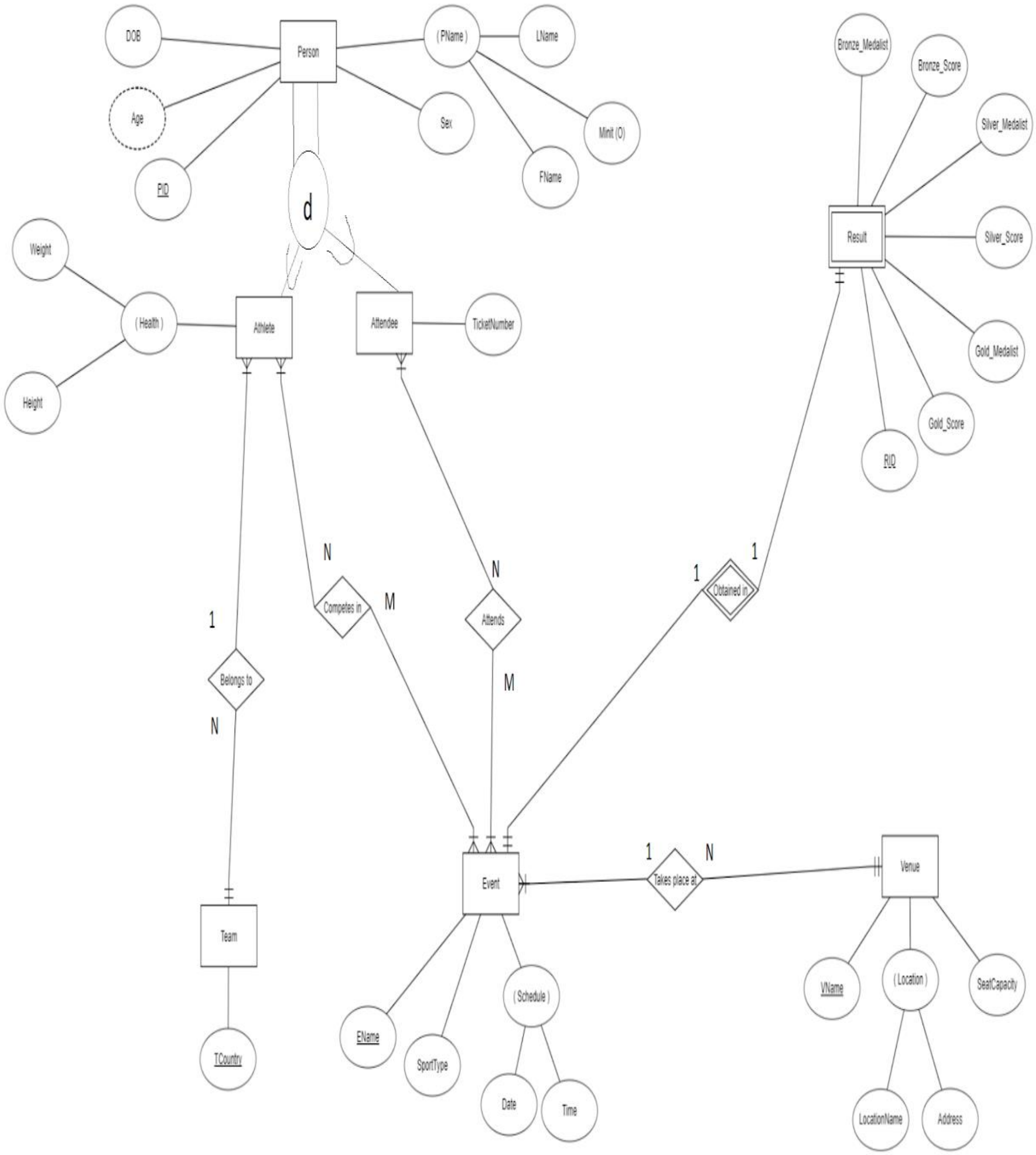
2. **Athlete competes in Event: An athlete can compete in one or more events at the Olympics. For example, Usain Bolt will compete in the 100M and 200M Men Finals. The events are participated by numerous athletes who compete with one another. For example, Adam Peaty, Cody Miller, and Yasuhiro Kousuke are some of the athletes who will compete in the 100M Breaststroke Men Finals. This is a many-to-many (N:M) relationship.**

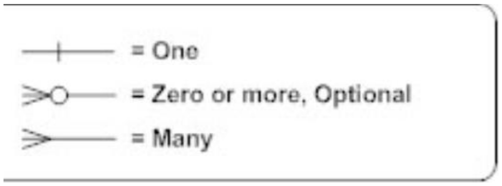
3. **Attendees attend Events: Attendees (i.e. the general public), to the Olympics can attend one or more events while an event is attended by many attendees. This is a many-to-many (N:M) relationship.**

4. **Event takes place at Venue: An event is located at one and only one venue at the Olympics while a venue hosts numerous events. For example, Sabre Team Men Semifinals and 48-55KG Wrestling Women Finals takes place at Makuhari Messe Convention Center. This is a one-to-many relationship (1:N).**

5. **Result obtained in Event: The result is the outcome of the event. Each event can either have a numeric or time-based score. The result also includes the top 3 highest scoring teams or individual athletes, resulting in the gold, silver, and bronze medalists. Each event only has one result and a result belongs to only one event. This is a one-to-one (1:1) relationship.**

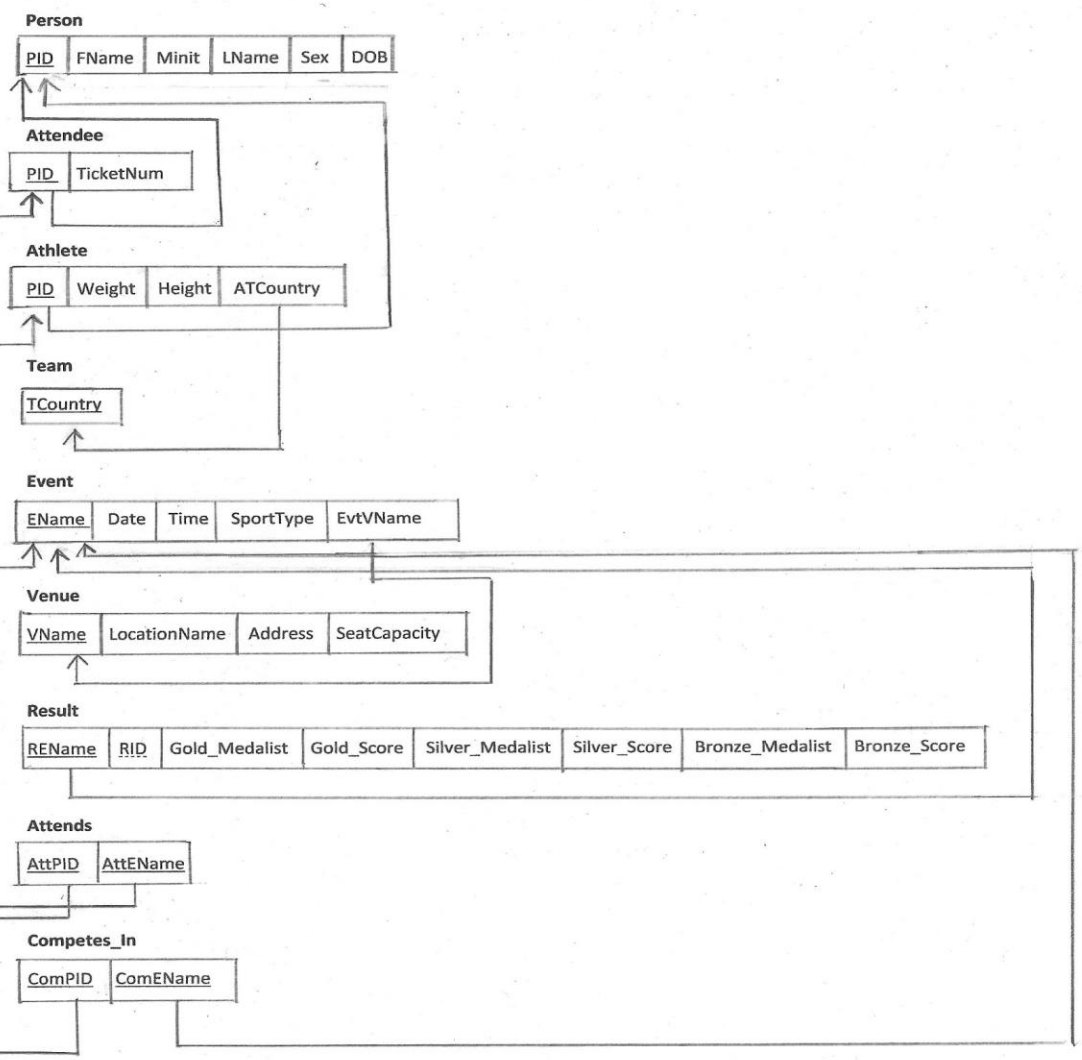
EER Model


Person
DOB
Age
PID
Sex
(PName)
LName
Minit (O)
FName
d
Athlete
Attendee
Weight
(Health)
Height
TicketNumber
Result
Bronze_Medalist
Bronze_Score
Silver_Medalist
Silver_Score
Gold_Medalist
Gold_Score
RID
N
Competes in
M
N
Attends
M
1
Obtained in
1
1
Belongs to
N
Team
TCountry
Event
EName
SportType
(Schedule)
Date
Time
1
Takes place at
N
Venue
VName
(Location)
SeatCapacity
LocationName
Address



(All entities have mandatory participation)

Relational Schema



Person

PID	FName	Minit	LName	Sex	DOB

Attendee

PID	TicketNum

Athlete

PID	Weight	Height	ATCountry

Team

TCountry

Event

EName	Date	Time	SportType	EvtVName

Venue

VName	LocationName	Address	SeatCapacity

Result

REName	RID	Gold_Medalist	Gold_Score	Silver_Medalist	Silver_Score	Bronze_Medalist	Bronze_Score

Attends

AttPID	AttEName

Competes_In

ComPID	ComEName

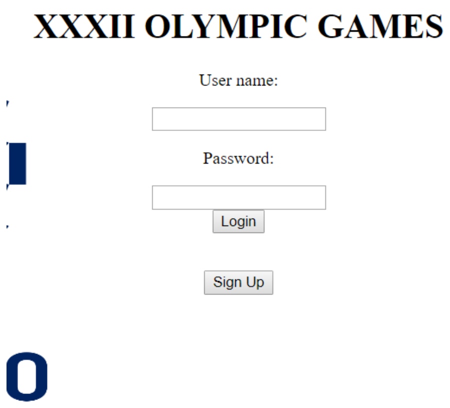
Design Decisions Summary

We decided a Person can be either an Athlete or Attendee, but not both (disjoint specialization). We used Option 8A to map the schema for Person so we can maintain the superclass Person and assign the primary key of Person (PID) to both Athlete and Attendee since PID itself can never be null. Athlete is assigned the foreign key, ATCountry from Team whose primary key is TCountry since an athlete can only belong to one team. Event is assigned the foreign key EvtVname from Venue whose primary key is Vname; this represents how each event takes place at a specific venue. Result is a weak entity because its existence is dependent on Event. Once an event has concluded, the results (scores, medalists) are created; thus, Results' primary key is a combination REName (Ename is the primary key of event) and the partial key RID. In addition, for the many-to -many relationships, AttPID (PID is the primary key of Attendee) and AttEname (EName is the primary key of Event) form the primary key of Attends while ComPID (PID is the primary key of Athlete) and ComEname (EName is the primary key of Event) form the primary key of Competes_In.

How to use the website

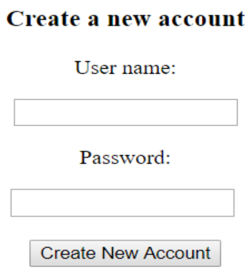
Link to the website: 73.76.181.77/olympics

This is the login screen shown on the homepage:

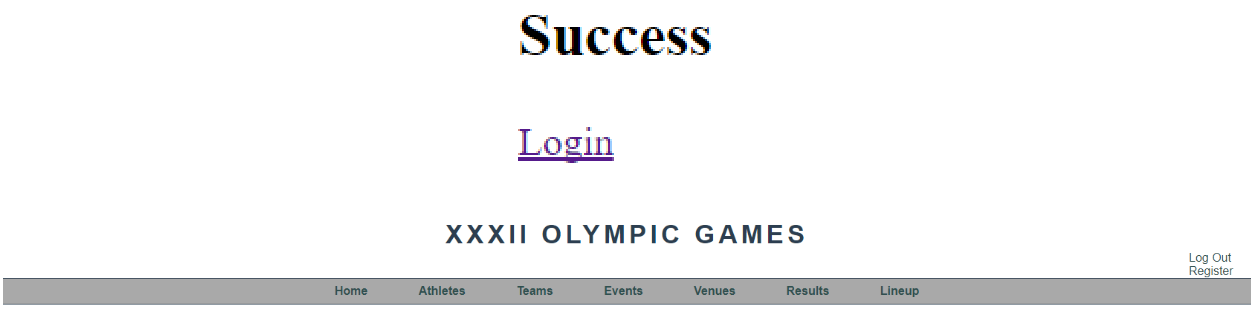


XXXII OLYMPIC GAMES

The two default users for the database are Admin, who will login with the username/password combination admin/admin and Guest, who will login with the username/password combination guest/guest. In order to create new guest accounts, click Sign Up. You will be directed to a page where you must fill in a new User Name and Password. Once you have completed, click Create New Account and you will be redirected to a Success page.



Create a new account



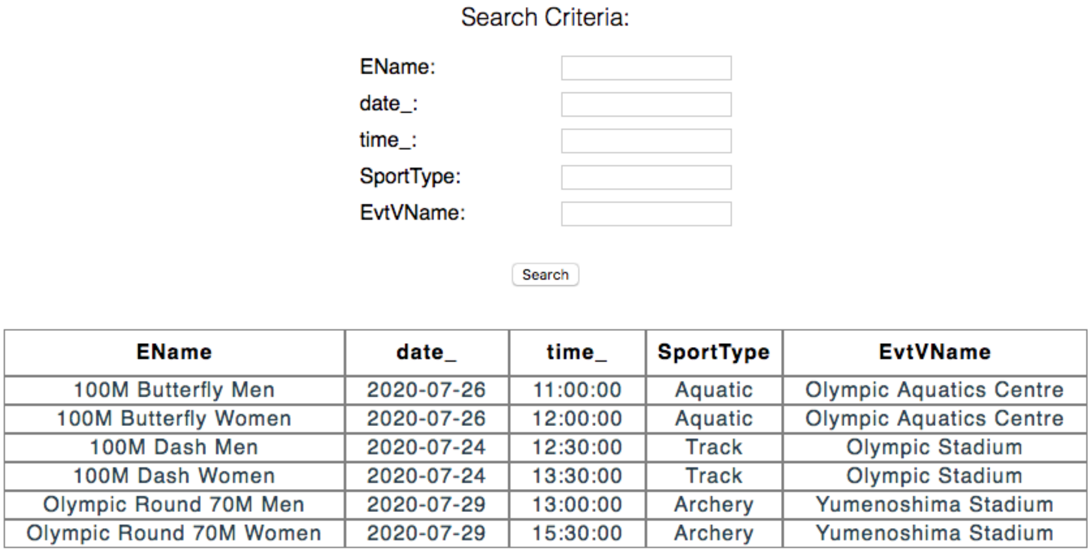
Success

Login

XXXII OLYMPIC GAMES

You can choose to login by clicking Login on the Success page or on the Home Page. Once a user is logged in, they can view the tables by clicking the different tabs (Athletes, Teams, Events, etc). To log out, simply click Log Out on the upper right hand corner. You can also create new guest accounts by clicking Register on the upper right hand corner.

All users are able to search by simply entering the data they want. It is not required to fill out all fields (attributes) to complete the search, that only specifies it further. A search may be completed with just one entity.



Search Criteria:

EName:

date_:

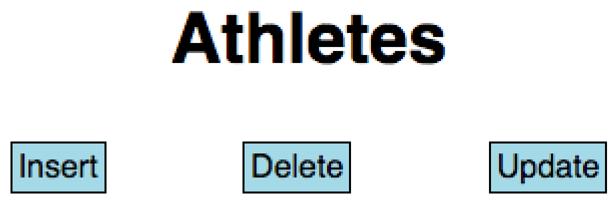
time_:

SportType:

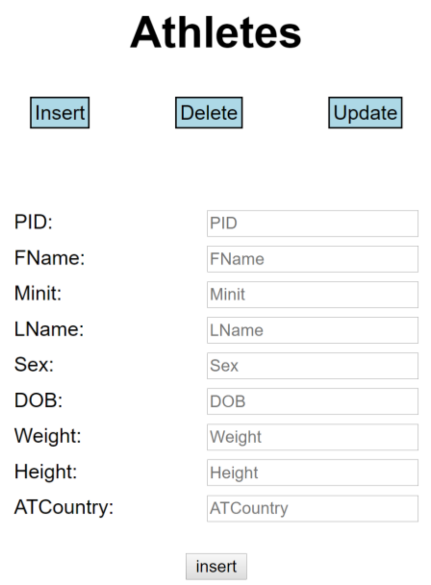
EvtVName:

EName	date_	time_	SportType	EvtVName
100M Butterfly Men	2020-07-26	11:00:00	Aquatic	Olympic Aquatics Centre
100M Butterfly Women	2020-07-26	12:00:00	Aquatic	Olympic Aquatics Centre
100M Dash Men	2020-07-24	12:30:00	Track	Olympic Stadium
100M Dash Women	2020-07-24	13:30:00	Track	Olympic Stadium
Olympic Round 70M Men	2020-07-29	13:00:00	Archery	Yumenoshima Stadium
Olympic Round 70M Women	2020-07-29	15:30:00	Archery	Yumenoshima Stadium

If the user has logged in as an admin, then they will also have the ability to insert, update, and delete rows by simply selecting the following buttons:

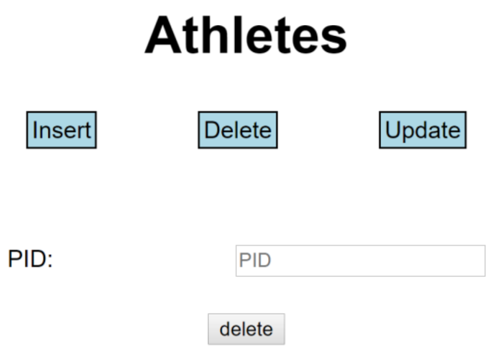

Athletes

To insert a row, the admin is required to enter values for all the fields (the only exception in the entire database is the middle initial (Minit) for Athletes, which is optional) and click the Insert button below.


Athletes

PID: PID
FName: FName
Minit: Minit
LName: LName
Sex: Sex
DOB: DOB
Weight: Weight
Height: Height
ATCountry: ATCountry

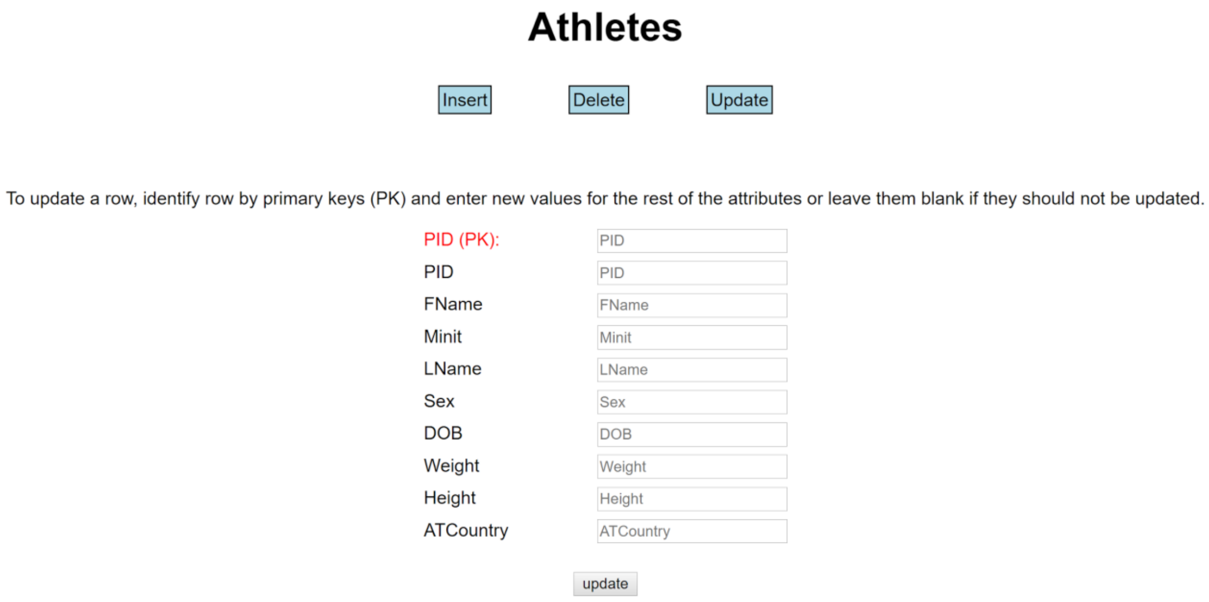
To delete a row, the admin must enter the primary key that corresponds to the row they wish to delete and click the Delete button below.


Athletes

PID: PID

To update a row, the admin must enter the primary key that corresponds to the row they wish to update in the text box that is highlighted red. The admin can then enter

new values for fields they wish to update or leave them blank if they should not be updated. Once they are finished, the admin must click the Update button below.



Athletes

To update a row, identify row by primary keys (PK) and enter new values for the rest of the attributes or leave them blank if they should not be updated.

PID (PK):	PID
PID	PID
FName	FName
Minit	Minit
LName	LName
Sex	Sex
DOB	DOB
Weight	Weight
Height	Height
ATCountry	ATCountry

Queries

```
CREATE DATABASE /*!32312 IF NOT EXISTS*/`Olympic_Games_Tokyo_2020_XXXII` /*!40100 DEFAULT
CHARACTER SET latin1 */;

USE Olympic_Games_Tokyo_2020_XXXII;

DROP TABLE IF EXISTS PERSON;

CREATE TABLE PERSON
(PID int(8) primary key,
FName varchar(20) not null,
Minit char(1) default null,
LName varchar(20) not null,
Sex char(1) not null,
DOB date not null
);

INSERT INTO Person(PID, FName, Minit, LName, Sex, DOB) VALUES
#Attendees
(114953, 'Isabelle', null, 'Charafeddine', 'F', '1996-03-05'),
(147364, 'Jeremy', null, 'Le', 'M', '1997-01-27'),
(173492, 'Benqiang', null, 'Lin', 'M', '1997-08-16'),
(158230, 'Brandon', null, 'Nguyen', 'M', '1995-06-30'),
(135924, 'Patricia', 'A', 'Sieng', 'F', '1996-11-20'),
(163729, 'Giulia', null, 'Toti', 'F', '1987-03-10'),
#100M Dash
#Men
(127693,'Usain', 'L', 'Bolt', 'M', '1986-08-21'), #Gold TScore: 9.81
(122241, 'Justin', null, 'Gatlin', 'M', '1982-02-10'), #Silver TScore: 9.89
(127431, 'Andre', 'D', 'Grasse', 'M', '1994-11-10'), #Bronze TScore: 9.91
#Women
(133590, 'Elaine', null, 'Thompson', 'F', '1992-06-28'), #Gold TScore: 10.71
(136821, 'Tori', null, 'Bowie', 'F', '1990-08-27'), #Silver TScore: 10.83
(137421, 'Dafne', null, 'Schippers', 'F', '1992-06-15'), #Bronze TScore: 10.96
#100M Butterfly
```

```
#Men
(137730, 'Michael', 'F', 'Phelps', 'M', '1986-06-30'), #Gold TScore: 1:53:36
(137621, 'Masato', null, 'Sakai', 'M', '1995-06-06'), #Silver TScore: 1:53:40
(137890, 'Tamas', null, 'Kenderesi', 'M', '1996-12-13'), #Bronze TScore: 1:53.62
#Women
(136554, 'Mireia','B', 'Garcia', 'F', '1990-11-10'), #Gold TScore 1:53:62
(136512, 'Madeline', null, 'Groves', 'F', '1995-05-25'), #Silver TScore: 2:04:85
(136735, 'Natsumi', null, 'Hoshi', 'F', '1990-08-20'), #Bronze TScore: 2:05:20
#Olympic Round 70M (Archery)
#Men
(141100, 'Ku', 'B', 'Chan','M', '1993-01-31'), #Gold NScore: 149
(127602, 'Jean', 'C', 'Valladont','M', '1989-03-20'), #Silver NScore: 141
(199693, 'Brady', 'L', 'Ellison', 'M', '1988-10-27'), #Bronze NScore: 117
#Women
(185673, 'Chang', 'H', 'Jin', 'F', '1987-05-13'), #Gold NScore: 115
(177249, 'Lisa', null, 'Unruh', 'F', '1988-04-12'), #Silver NScore: 109
(170019, 'Ki', 'B', 'Bae',  'F', '1988-02-20'); #Bronze NScore: 137

CREATE VIEW P AS
SELECT *
FROM PERSON;

SELECT *
FROM P;

#Additional Query
/*SELECT Fname, Minit, LName
FROM PERSON
WHERE Sex = 'F' AND DOB > '1990-01-01';*/
DROP TABLE IF  EXISTS ATTENDEE;

CREATE TABLE ATTENDEE
(PID integer,
TicketNumber integer,
constraint ATTENDEE_PK primary key(PID),
constraint ATTENDEE_FK foreign key(PID) references Person(PID)
);

INSERT INTO Attendee (PID, TicketNumber) VALUES
#Attendees
(114953, '10'), #Isabelle
(147364, '20'), #Jeremy
(173492, '30'), #Benqiang
(158230, '40'), #Brandon
(135924, '50'), #Patricia
(163729, '60'); #Giulia

CREATE VIEW ATD AS
SELECT *
FROM ATTENDEE;

SELECT *
FROM ATD;

#Additional Query
/*SELECT FName, LName
FROM Attendee, Person
WHERE Attendee.PID = Person.PID AND Sex = 'M';*/

DROP TABLE IF EXISTS TEAM;

CREATE TABLE TEAM
(TCountry varchar(20) primary key
);

INSERT INTO TEAM (TCountry) VALUES
```

```
#Attendees
('Jamaica'),
('America'),
('Canada'),
('Netherlands'),
('Japan'),
('Hungary'),
('Spain'),
('Australia'),
('Korea'),
('France'),
('Germany');

CREATE VIEW T AS
SELECT *
FROM TEAM;

SELECT*
FROM T;

DROP TABLE IF EXISTS ATHLETE;

CREATE TABLE ATHLETE
(PID integer,
Weight integer not null,
Height integer not null,
ATCountry varchar(20) not null,
Constraint pk_Att primary key(PID),
Constraint fk_Att foreign key(ATCountry) references TEAM(TCountry),
Constraint fk_Att2 foreign key(PID) references PERSON(PID)
);

INSERT INTO Athlete (PID, Weight, Height, ATCountry) VALUES
```

```
#100M Dash
#Men
(127693, 207, 195, 'Jamaica'), #Usain
(122241, 175, 185, 'America'), #Justin
(127431, 154, 176, 'Canada'), #Andre
#Women
(133590, 126, 167, 'Jamaica'), #Elaine
(136821, 121, 175, 'America'), #Tori
(137421, 150, 179, 'Netherlands'), #Dafne
#100M Butterfly
#Men
(137730, 194, 193, 'America'), #Michael
(137621, 170, 181, 'Japan'), #Masato
(137890, 165, 186, 'Hungary'),  #Tamas
#Women
(136554, 132, 170, 'Spain'), #Mireia
(136512, 145, 179, 'Australia'), #Madeline
(136735, 119, 164, 'Japan'), #Natsumi
#Olympic Round 70M (Archery)
(141100, 185, 181, 'Korea'), #Ku
(127602, 183, 180, 'France'), #Jean
(199693, 190, 183, 'America'), #Brady
#Olympic Round 70M (Archery)
(185673, 110, 158, 'Korea'), #Chang
(177249, 159,  178, 'Germany'), #Lisa
(170019, 126, 167, 'Korea'); #Ki

CREATE VIEW A AS
SELECT *
FROM ATHLETE;

SELECT *
FROM A;
```

```
#Additional Query
/*SELECT FName, Minit, LName
FROM Athlete, Person
WHERE Athlete.PID = Person.PID AND ATCountry = 'America' AND Sex = 'M';*/

DROP TABLE IF EXISTS VENUE;

CREATE TABLE VENUE
(VName varchar(30),
LocationName varchar(20) not null,
Address varchar(50) not null,
SeatCapacity integer not null,
Constraint pk_ven primary key(VName)
);

INSERT INTO VENUE VALUES
('Olympic Stadium', 'Heritage Zone', '10-2 Kasumigaokamachi, Shinjuku, Tokyo', 60000), #100M Dash
('Olympic Aquatics Centre', 'Tokyo Bay Zone', '1 Chrome-16-2 Kaigan, Minato, Tokyo', 18000), #100M Butterfly
('Yumenoshima Stadium', 'Tokyo Bay Zone', '1 Chrome-16-2 Kaigan, Koto, Tokyo', 6000), #Olympic Round 70M
('Tokyo Metropolitan Gymnasium', 'Heritage Zone', '1 Chrome-17-1 Sendagaya, Shibuya, Tokyo', 10000), #Additional location
('Nippon Budokan', 'Heritage Zone', '2-3 Kitanomarukoen, Chiyoda, Tokyo', 12000); #Additional location

CREATE TABLE V AS
SELECT *
FROM VENUE;

SELECT *
FROM V;

#Additional Query
/*SELECT VName, Address, SeatCapacity
FROM VENUE
```

```
WHERE LocationName = 'Tokyo Bay Zone' AND SeatCapacity > 5000;*/

DROP TABLE IF EXISTS EVENT_;

CREATE TABLE EVENT_
(EName varchar(30) primary key,
date_ date not null,
time_ time not null,
SportType varchar(20) not null,
EvtVName varchar(30) not null,
Constraint Evt_fk foreign key(EvtVName) references VENUE(VName));

INSERT INTO Event_ (Ename, date_, time_, SportType, EvtVName) values
('100M Dash Men', '2020-07-24', '12:30:00', 'Track', 'Olympic Stadium'),
('100M Dash Women ', '2020-07-24', '13:30:00', 'Track', 'Olympic Stadium'),
('100M Butterfly Men', '2020-07-26', '11:00:00', 'Aquatic', 'Olympic Aquatics Centre'),
('100M Butterfly Women', '2020-07-26', '12:00:00', 'Aquatic', 'Olympic Aquatics Centre'),
('Olympic Round 70M Men', '2020-07-29', '13:00:00', 'Archery', 'Yumenoshima Stadium'),
('Olympic Round 70M Women', '2020-07-29', '15:30:00', 'Archery', 'Yumenoshima Stadium');

CREATE VIEW E AS
SELECT *
FROM EVENT_;

SELECT *
FROM E;

#Additional Query
/*SELECT EName, date_, time_, SportType
FROM EVENT_
WHERE EvtVName = 'Olympic Stadium';*/

CREATE TABLE RESULT
```

```
(R_EName varchar(30),
RID integer,
Gold_Medalist varchar(50) not null,
Gold_Score varchar(10) not null,
Silver_Medalist varchar(50) not null,
Silver_Score varchar(10) not null,
Bronze_Medalist varchar(50) not null,
Bronze_Score varchar(10) not null,
constraint pk_RE primary key(R_EName, RID),
constraint fk_RE foreign key(R_EName) references EVENT_(EName)
);

INSERT INTO RESULT (R_Ename, RID, Gold_Medalist, Gold_Score, Silver_Medalist, Silver_Score, Bronze_Medalist, Bronze_Score)VALUES
('100M Dash Men', 223417, 'Usain L Bolt', '9.81', 'Justin Gatlin', '9.89', 'Andre D Grasse', '9.91'),
('100M Dash Women', 278452, 'Elaine Thompson', '10.71', 'Tori Bowie', '10.83', 'Dafne Schippers', '10.96'),
('100M Butterfly Men', 216783, 'Michael F Phelps', '1:53:36', 'Makoto Sakai', '1:53:40', 'Tamas Kenderesi', '1:53:62'),
('100M Butterfly Women', 285437, 'Mireia B Garcia', '1:53:62', 'Madeline Groves', '2:04:85', 'Natsumi Hoshi', '2:05:20'),
('Olympic Round 70M Men', 264123, 'Ku B Chan', '149', 'Jean C Valladont', '141', 'Brady L Ellison', '117'),
('Olympic Round 70M Women', 225819, 'Chang H Jin', '115', 'Lisa Unruh', '109', 'Ki B Bae', '137');

CREATE VIEW R AS
SELECT *
FROM RESULT;

SELECT *
FROM R;

#Additional Query
/*SELECT *
FROM RESULT
WHERE R_EName = '100M Dash Men' OR '100M Dash Women';*/
```

```
DROP TABLE IF EXISTS Attends;

CREATE TABLE Attends
(AttPID integer,
AttEName varchar(30),
constraint ATT_PK primary key(AttPID, AttEName),
constraint ATT_FK foreign key(AttPID) references ATTENDEE(PID),
constraint ATT_FK_2 foreign key(AttEname) references Event_(EName)
);

INSERT INTO Attends VALUES
(114953, '100M Dash Men'), #Isabelle
(147364, '100M Dash Women'), #Jeremy
(173492, '100M Butterfly Men'), #Benqiang
(158230, '100M Butterfly Women'), #Brandon
(135924, 'Olympic Round 70M Men'), #Patricia
(163729, 'Olympic Round 70M Women'); #Giulia

CREATE VIEW ATT AS
SELECT *
FROM  Attends;

SELECT  *
FROM ATT;

#Additional Query
/*SELECT AttPID, Fname, LName, EName
FROM  PERSON, attends, EVENT_
WHERE EName = AttEname AND PID = AttPID;*/
DROP TABLE IF EXISTS Competes_In;

CREATE TABLE Competes_In
```

```
(ComPID integer,
ComEName varchar(30),
constraint COM_PK primary key(ComPID, ComEName),
constraint COM_FK foreign key(ComPID) references ATHLETE(PID),
constraint COMP_FK_2 foreign key(ComEname) references EVENT_(EName)
);

INSERT INTO Competes_In VALUES
(127693, '100M Dash Men'), #Usain
(122241, '100M Dash Men'), #Justin
(127431, '100M Dash Men'), #Andre
(133590, '100M Dash Women'), #Elaine
(136821, '100M Dash Women'), #Tori
(137421,'100M Dash Men' ), #Dafne
(137730, '100M Butterfly Men'), #Micheal
(137621, '100M Butterfly Men'), #Makoto
(137890, '100M Butterfly Men'), #Tamas
(136554, '100M Butterfly Women'), #Miera
(136512, '100M Butterfly Women'), #Madeline
(136735, '100M Butterfly Women'), #Natsumi
(141100, 'Olympic Round 70M Men'), #Ku
(127602, 'Olympic Round 70M Men'), #Jean
(199693, 'Olympic Round 70M Men'), #Brady
(185673, 'Olympic Round 70M Women'), #Chang
(177249, 'Olympic Round 70M Women'), #Lisa
(170019, 'Olympic Round 70M Women'); #Ki

CREATE VIEW C AS
SELECT *
FROM Competes_In;

SELECT *
FROM C;
```

```
#Additional Query
/*SELECT ComPID, FName, LName, Sex, AtCountry, ComEName
FROM Competes_In, Person, Athlete
WHERE Person.PID = ComPID AND Athlete.PID = ComPID;*/
```