Project Group 4

Amy Mummert / Joel Vrieze

OSU CS340 Database Project

URL http://flip3.engr.oregonstate.edu:6484/

Baton Twirling Competition Entry Form Database Project Proposal

Executive Summary - Feedback and Changes to Project

The Baton Twirling Competition Entry Form Database provides an easy and simple way to electronically enter into competition events. Competition facilitators can add, update, delete, and search for competitions, athletes, and within athletes events. They can also add and delete events, divisions, event levels, and teams. Athletes, Teams, and Coaches can also use this database to enter into events.

Step 1 - Created overview, database outline, and ERD:

During the step one process we created our project overview, outline, and ERD. Our original plan was to create 9 tables(Competitions, Locations, Events, Athletes, Coaches, Teams, EventLevels, and Divisions) and one M:N relationship table called Athletes_Events. We eliminated the coaches table and combined it with the teams table and renamed the Level entity to EventLevels(now a category table). We also eliminated the Locations table and combined it with Competitions. We made these changes to simplify our database design. We also added athleteDOB to the Athletes entity so the database could self calculate athleteAge. Last, we updated the ERD to reflect any changes we made.

Step 2 - Updates to all steps + Schema and DDL:

During the step two process we updated all of the above steps and created our Schema and DDL. We updated some attributes to reflect when NULL/NOT NULL should be used. We then made our ERD with draw.io instead of mysql workbench. This change allowed us to improve our ERD simplicity and display both PKs and FKs. Our ERD reflects all tables including the M:N relationship table with all relationships displayed with lines to each table.

Step 3 - Updates to all steps + Design HTML Interface and DML SQL:

During the step three process we updated all of the above steps and created our HTML interface and Data Manipulation Queries(DDL). We clarified our naming technique as camelCase as there were some discrepancies with naming within our ERD and DDL. We used PhpMyAdmin to build our schema and clearly defined entities and attributes. We switched the order of athleteDOB and athleteAge which now allows athleteDOB to calculate athleteAge in that order. We made several changes to the search/filter queries to explain how a user is allowed to search that table.

Step 4 - Updates to all steps + Implement CRUD for One Entity:

During the step 4 process we implemented CRUD for one Entity - Competitions. At step four we decided to use express handlebars with node.js. We did this not only for experience but also because we felt handlebars would assist us in creating a nicely programmed webapp. We implemented CRUD steps ADD/DELETE/UPDATE and Search to Competitions. We also made updates to our DDL and DML accordingly. While completing this step we also implemented the index.hbs and main.js files. We eliminated the entire delete section from our entities and added it as a button in the browse table. This eliminated the need for a user to scroll to the bottom of the page to delete something. We fixed our search queries so they returned the right information. At the end of this step we had add, delete, and search working for competitions, but had difficulty implementing update_competition.

Step 5 - Implemented most remaining entities:

During the step 5 process we worked on implementing our remaining entities. We spent a lot of time troubleshooting errors and implementing express handlebars and ajax. By the end of this step we had 4 out of 7 entities working. The entities were EventLevels, Events, Divisions, Teams with ADD/DELETE and Search implementation. Competitions and Athletes ADD/DELETE and Search all worked, but again UPDATE was giving us trouble.

Step 6 - Finished Project - We continued with implementing our M:N relationship, getting the UPDATE CRUD step working and finished up the project.

Project Title

Baton Twirling Competition Entry Form Database

Team Name

Apple Jacks

Authors: Amy Mummert and Joel Vrieze – Group 4

Overview - Briefly describe the problem that a website with a database backend will solve.

This proposal outlines the Baton Twirling Competition Entry Form Database created by Amy Mummert and Joel Vrieze. The Baton Twirling Competition Entry Form Database (BTCE) creates an entry form for baton twirling competitions for approximately 52 competitions per year and hundreds of events. A competition includes events that baton twirling athletes can register for and compete in. Athletes who enter in a competition can register to enter one or more events within that competition. Events include, but are not limited to, basic march, military march, parade march, presentation, 2-baton, 3-baton, solo, strut and artistic twirl. The database includes six entity tables, one category table and six relationships between tables. The tables are Competitions, Events, Teams, Athletes, and Divisions. The category table is named EventLevels. The database has one transition table between Events and Athletes, named Athletes_Events. This proposal includes an overview, a database outline, and an ERD diagram.

A database driven website for baton twirling competitions will record event entries by athletes at competitions. The database can hold thousands of records which can easily be accessed by Competition staff. Approximately 400 athletes compete in a single competition at one time, but this number is not set in stone because there is no cutoff of entries per competition. Competitions last from one day to a few days at a time, possibly more. Competitions for baton twirling can be at many different locations but the location cannot have more than one competition at a time. If a location has ten competitions a year then approximately 10,000 athletes compete in those competitions. It is possible for a location to have less competitions per year and many competitions occurring per year at different locations. This statistic will become more accurate when taking an average after a few years.

Databases are essential to many businesses and large corporations. Databases store data in a collection that is easily accessible to a business. A relational database, stores data in tables and with those tables creates relationships. Relational databases (one type of back-end database), solve several problems that a website without a back-end database might have. By eliminating paper entry forms and creating the baton twirling entry form database, paper entries won't get lost or misplaced and event entries will be well organized. Competition staff can easily see which athletes signed up for which events and records will be retained in the database.

A relational database for baton twirling solves several issues. Data redundancy or duplication of data becomes non-existent within a relational database. In our case, duplicate entries for one event by an athlete won't be an issue with a relational database. Data accuracy, which is very important to businesses, remains consistent by updating all instances of a table and its associated data. With a database for baton twirling entries will be consistent and accurate and competition staff won't have to worry about reading someone's handwriting. Data recovery is also an issue that a relational database solves. Data for baton twirling can be backed up and recovered easily. Data integrity is solved by only allowing certain competition staff to access the database data. With this, a baton twirling database can maintain confidentiality and integrity and improve the availability of data. Querying becomes quicker and faster and uses SQL (Structured Query Language) to perform CRUD operations (Create, Read, Update, Delete). This allows competition staff to focus more on set-up and organization, for example, than entering all entries by hand.

Database Outline

Competitions: Records competition details including name, date, location, and start time.

- competitionID: int, PK, NN, unique, auto_increment
- competitionName: VARCHAR(255), NN
- date: DATE
- startTime: TIME
- locationName: VARCHAR(255), NN
- locationAddress: VARCHAR(255), NN
- locationPhone: VARCHAR(15), Null
- Relationship: A 1:M relationship between Competitions and Events is implemented with competitionID as an FK inside events. For each competition there are one to many events.

Events: Records the event details for a given event type with division and eventlevel included. Events include, but are not limited to, basic march, military march, parade march, presentation, 2-baton, 3-baton, solo, strut and artistic twirl.

- eventID: int, PK, NN, unique, auto_increment
- competitionID: int, FK, Null
- divisionID: int, FK, NN
- eventlevelID: int, FK, NN
- eventName: VARCHAR(255), NN
- Relationship: A M:M relationship between Events and Athletes is implemented with a transition table, where there can be 1:M relationship between Athletes and Events where each athlete could have zero to many event entries and a 1:M between an Events and Athletes where each event could have zero to many athletes.

Athletes: Records details for each athlete in the database including athlete name, athlete address, athlete phone number and date of birth which will determine the athletes age on or before 8/31 of the current year.

- athleteID: int, PK, NN, unique, auto_increment
- teamID: int, FK, NULL
- divisionID, int, FK, NULL
- athleteName: VARCHAR(255), NN
- athleteAddress: VARCHAR(255) NN
- athletePhone: VARCHAR(15), NN
- athleteEmail: VARCHAR(255)
- athleteDOB: date,NN
- athleteAge: int, NN (age athlete will be on or before 8/31 of current year based on athleteDOB)
- Relationship: A M:M defined in Events and a M:1 defined in Divisions.

Teams: Records the team details for each team in the database including team name and coach.

- teamID: int, PK, NN,unique, auto_increment
- teamName: VARCHAR(255), NN
- coachName: VARCHAR(255), NN
- coachPhone: VARCHAR(15), NN
- coachEmail: VARCHAR(255), NN

- Relationship: A 1:M relationship between Teams and Athletes is implemented with teamID as an FK inside Athletes. For each Team there has to be one or more athletes on the team. Each Athlete must be on a Team.

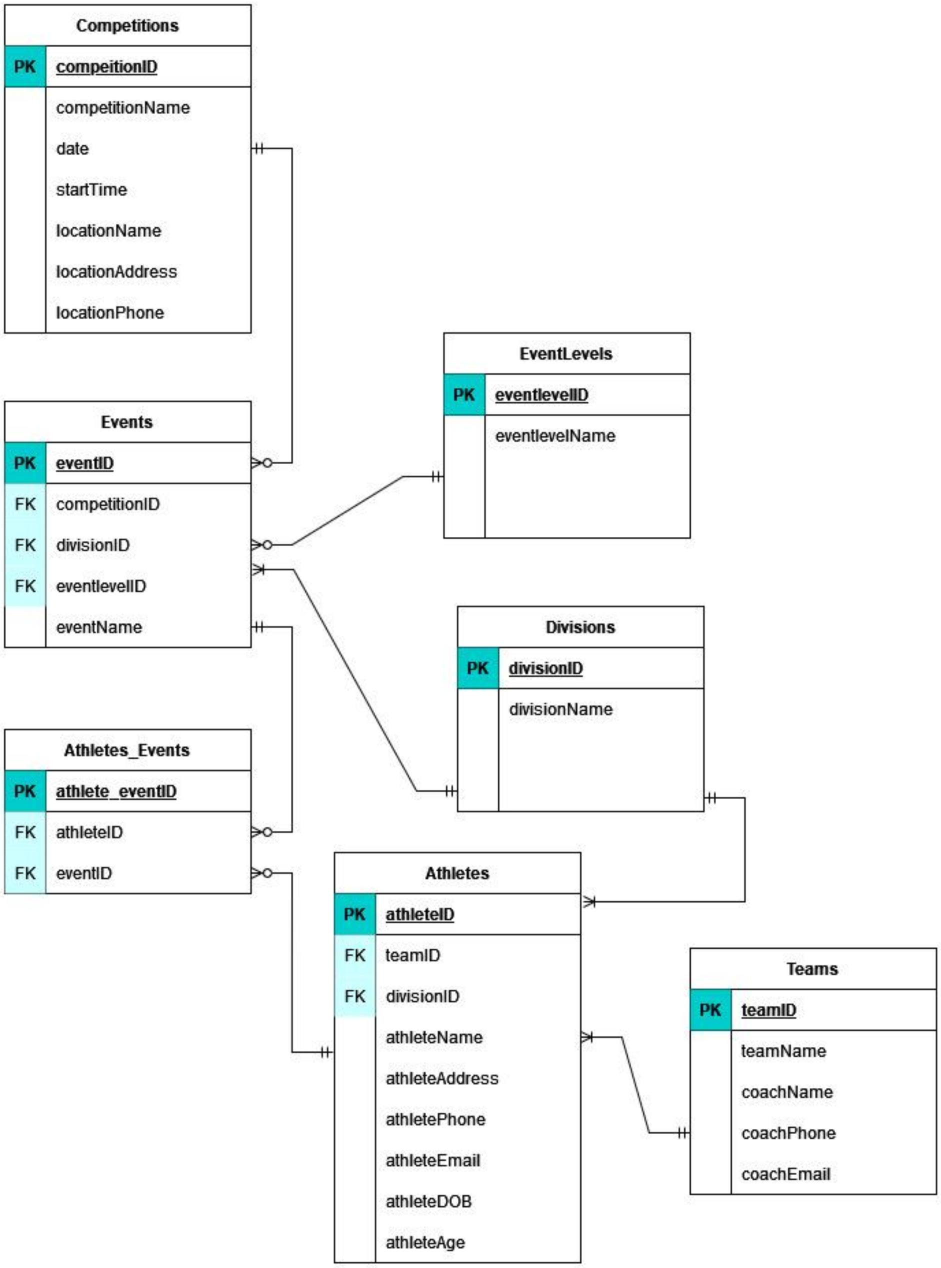
Divisions: Records details for the event and athletes division.

- divisonID: int, PK, NN, auto_increment
- divisonName: VARCHAR(255), NN
- Relationship: 1:M relationship between Divisions and Events which is implemented with divisionID as an FK inside Events. Each division can be assigned to zero to many events. 1:M relationship between Division and Athletes which is implemented with divisionID as an Fk inside Athletes. Each division can be assigned to zero to many Athletes.

EventLevels: A Category table for event levels. It Records details for each event's level detail in baton twirling.

- eventlevelID: int, PK, NN, auto_increment
- eventlevelName: VARCHAR(255), NN
- Relationship: 1:M relationship between EventLevels and Events is implemented with eventlevelID as an FK inside Events. Each level can be assigned to zero to many events.

Entity Relationship Diagram (ERD)



Competitions	
PK	compeitionID
	competitionName
	date
	startTime
	locationName
	locationAddress
	locationPhone

Events	
PK	eventID
FK	competitionID
FK	divisionID
FK	eventlevelID
	eventName

Athletes_Events	
PK	athlete_eventID
FK	athleteID
FK	eventID

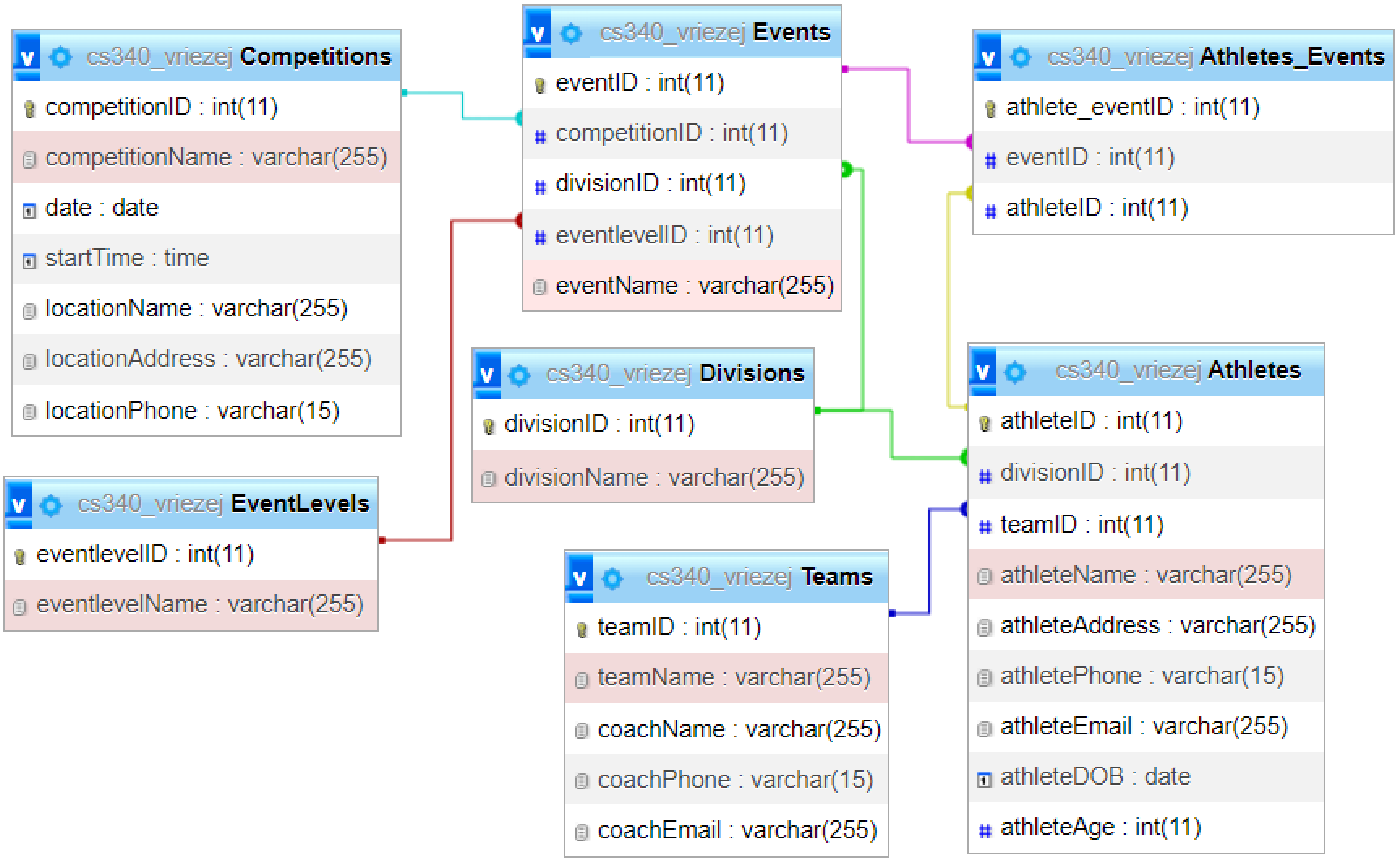
EventLevels	
PK	eventlevelID
	eventlevelName

Divisions	
PK	divisionID
	divisionName

Athletes	
PK	athleteID
FK	teamID
FK	divisionID
	athleteName
	athleteAddress
	athletePhone
	athleteEmail
	athleteDOB
	athleteAge

Teams	
PK	teamID
	teamName
	coachName
	coachPhone
	coachEmail

Schema



cs340_vriezej Competitions
competitionID : int(11)
competitionName : varchar(255)
date : date
startTime : time
locationName : varchar(255)
locationAddress : varchar(255)
locationPhone : varchar(15)

cs340_vriezej EventLevels
eventlevelID : int(11)
eventlevelName : varchar(255)

cs340_vriezej Events
eventID : int(11)
competitionID : int(11)
divisionID : int(11)
eventlevelID : int(11)
eventName : varchar(255)

cs340_vriezej Divisions
divisionID : int(11)
divisionName : varchar(255)

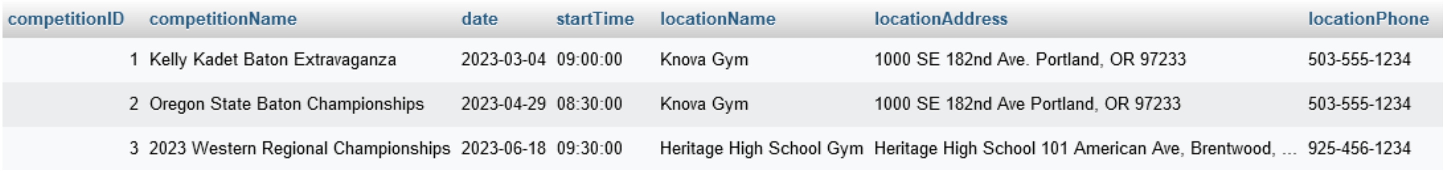
cs340_vriezej Teams
teamID : int(11)
teamName : varchar(255)
coachName : varchar(255)
coachPhone : varchar(15)
coachEmail : varchar(255)

cs340_vriezej Athletes_Events
athlete_eventID : int(11)
eventID : int(11)
athleteID : int(11)

cs340_vriezej Athletes
athleteID : int(11)
divisionID : int(11)
teamID : int(11)
athleteName : varchar(255)
athleteAddress : varchar(255)
athletePhone : varchar(15)
athleteEmail : varchar(255)
athleteDOB : date
athleteAge : int(11)

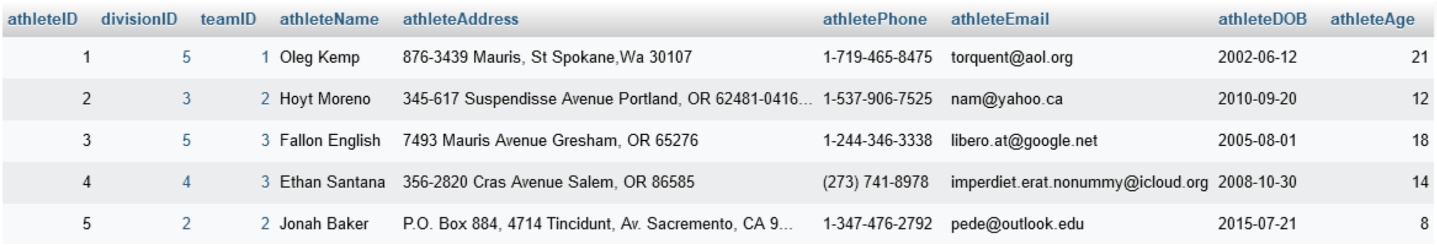
Example Data

Competitions:



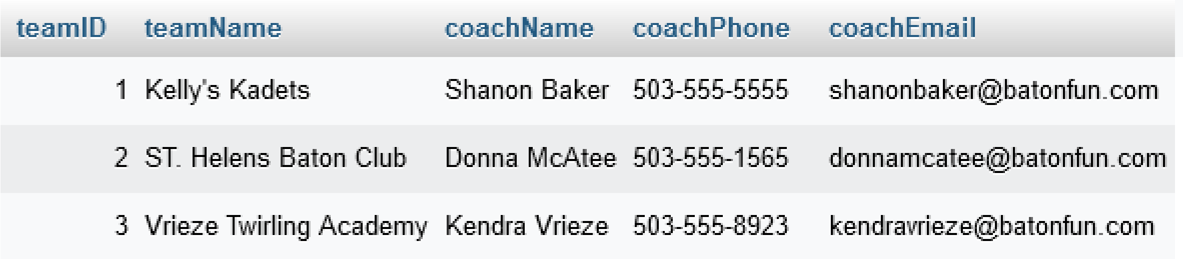
competitionID	competitionName	date	startTime	locationName	locationAddress	locationPhone
1	Kelly Kadet Baton Extravaganza	2023-03-04	09:00:00	Knova Gym	1000 SE 182nd Ave. Portland, OR 97233	503-555-1234
2	Oregon State Baton Championships	2023-04-29	08:30:00	Knova Gym	1000 SE 182nd Ave Portland, OR 97233	503-555-1234
3	2023 Western Regional Championships	2023-06-18	09:30:00	Heritage High School Gym	Heritage High School 101 American Ave, Brentwood, ...	925-456-1234

Athletes:



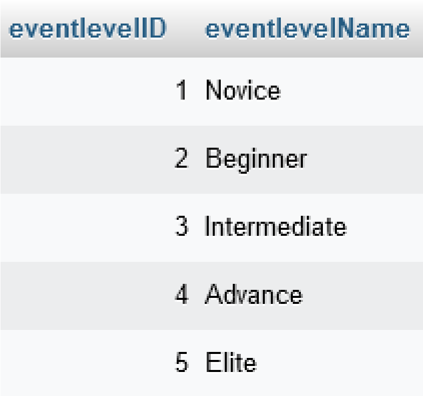
athleteID	divisionID	teamID	athleteName	athleteAddress	athletePhone	athleteEmail	athleteDOB	athleteAge
1	5	1	Oleg Kemp	876-3439 Mauris, St Spokane,Wa 30107	1-719-465-8475	torquent@aol.org	2002-06-12	21
2	3	2	Hoyt Moreno	345-617 Suspendisse Avenue Portland, OR 62481-0416...	1-537-906-7525	nam@yahoo.ca	2010-09-20	12
3	5	3	Fallon English	7493 Mauris Avenue Gresham, OR 65276	1-244-346-3338	libero.at@google.net	2005-08-01	18
4	4	3	Ethan Santana	356-2820 Cras Avenue Salem, OR 86585	(273) 741-8978	imperdiet.erat.nonummy@icloud.org	2008-10-30	14
5	2	2	Jonah Baker	P.O. Box 884, 4714 Tincidunt, Av. Sacremento, CA 9...	1-347-476-2792	pede@outlook.edu	2015-07-21	8

Teams:



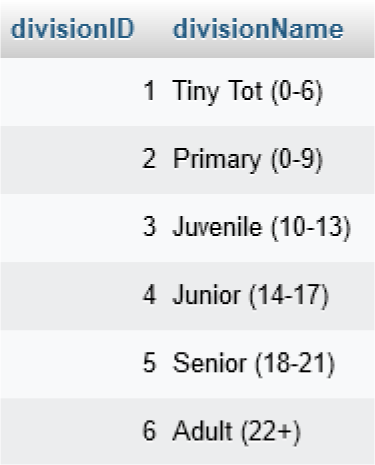
teamID	teamName	coachName	coachPhone	coachEmail
1	Kelly's Kadets	Shanon Baker	503-555-5555	shanonbaker@batonfun.com
2	ST. Helens Baton Club	Donna McAtee	503-555-1565	donnamcatee@batonfun.com
3	Vrieze Twirling Academy	Kendra Vrieze	503-555-8923	kendravrieze@batonfun.com

EventLevels:



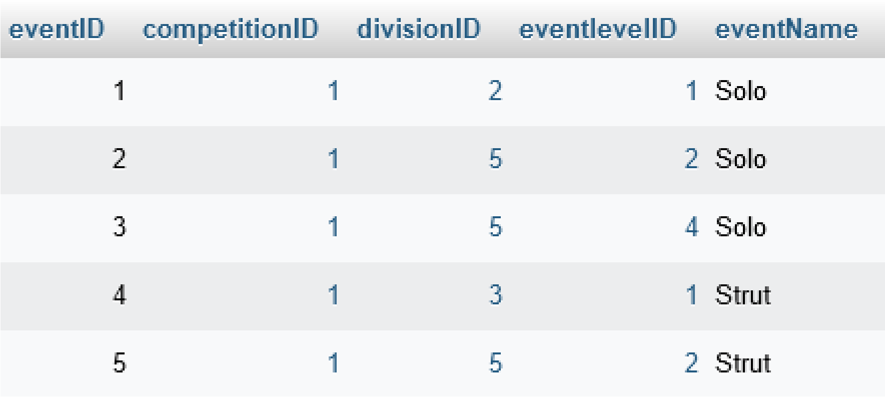
eventlevelID	eventlevelName
1	Novice
2	Beginner
3	Intermediate
4	Advance
5	Elite

Divisions:



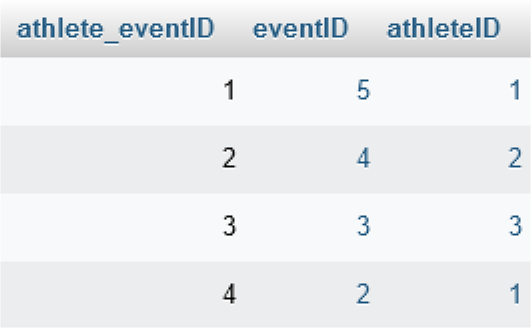
divisionID	divisionName
1	Tiny Tot (0-6)
2	Primary (0-9)
3	Juvenile (10-13)
4	Junior (14-17)
5	Senior (18-21)
6	Adult (22+)

Events:



eventID	competitionID	divisionID	eventlevelID	eventName
1	1	2	1	Solo
2	1	5	2	Solo
3	1	5	4	Solo
4	1	3	1	Strut
5	1	5	2	Strut

Athletes_Events:



athlete_eventID	eventID	athleteID
1	5	1
2	4	2
3	3	3
4	2	1

UI Screen Captures

Homepage - Index.hbs