Project Group 4
Amy Mummert / Joel Vrieze
OSU CS340 Database Project

Baton Twirling Competition Entry Form Database Project Proposal

URL

http://flip3.engr.oregonstate.edu:6484/ (link that is running forever)

http://flip3.engr.oregonstate.edu:4460/ (Joel's test work link) - Will have comments with changes

http://flip3.engr.oregonstate.edu:8013/ (Amy's test work link) - Will have comments withchanges

Actions Based On Feedback- Step 1 Draft

- Our overview now includes distinguishing features of Competition vs. Events.
- We added numerical values to the overview paragraph.
- We added types of events into the overview paragraph and in the Events description.
- We did not add our Intersection table to our outline because it is not needed in the outline.
- We changed the Level Entity to a category table and renamed it EventLevels.
- We left in the Team Entity because it needs to be a part of Athletes and eliminated the Coaches table and placed coach information into the Teams table.
- We updated the ERD to reflect accurate information and used draw.io to add FKs to the diagram.

Upgrades to the Draft version - Step 1 Draft

- We moved the overview paragraph for our database description on top of the paragraphs about a relational database design and how it can solve several issues.
- We removed levelID from Athletes as it is not needed in our database.
- We added a few more details to our overview.
- We added AthleteDOB to athletes to make database be able to self calculate AthleteAge which then will calculate DivisionID for an Athlete
- We eliminated the Locations table and the Coaches table, placing attributes into the Competition table and Teams table.
- Changed VARCHAR values to 255, 15 for phone numbers
- Added auto_increment to EventLevels
- Added athleteEmail to Athletes
- Updated ERD and schema to reflect changes

Upgrades and Actions Based on Feedback - Step 2 Draft / Final

- Updated varchar() data types to appropriate characters allowed number for the information being reflect
- Updated some attributes NULL or NOT NULL depending on if we wanted to allow or not allow something to be Cascade on delete or Set NULL on delete

- Removed LocationID attribute fromCompetitions in our database Outline since we had removed Locations entity from our design but forgot to remove that attribute for Competitions
- fixed in database Outline AthleteAddress in Athlete to only display varchar(255) as it's data type not both that and PostalAddress
- Added auto_increment to Athletes, Teams, Divisions and EventLevels ID attribute and removed unique if it was there since auto_increment makes each one unique.
- Updated ERD to reflect this change and also Updated ERD to have less detailed information so it is something to get the understanding without the schema which is below that.

Actions Based On Feedback - Step 3 Draft

- Athletes_Events is a intersection table so we chose to keep the underscore in both the Entity Name and athlete_eventID to signify this. EventLevels is a category table and so to follow all other naming conventions we left it as it.
- Since Athletes_Events is a transition table we chose to leave it out of our outline.
- We used PhpMyAdmin to build the schema and arranged lines as best we could so they wouldn't cross. We felt our lines in the schema clearly defined what was happening and did not make any changes.
- Changed wording on statistics for database in overview.
- Fixed athlete_eventID in ERD

Upgrades to the Project Step 3 Draft

- Switched AthleteDOB and AthleteAge in the ERD to reflect Schema order.

Actions Based On Feedback - Step 3 Final

- We upgraded our Search/Filters to explain how the user was allowed to search that table. Wrote queries to also search/filter and return the row of the table that the user was searching for.
- We also updated and changed the font of some of our pages to be more readable where we determined it would be appropriate to change.

Upgrades to the Project Step 3 - Step 3 Final

- We plan on changing the display titles of some tables to make them appear cleaner and have the title be more intuitive along with continuing to update CSS.

Actions Based On Feedback - Step 4 Draft

- Made some adjustments to the queries to accurately reflect some of the suggestions.

Upgrades to the Project Step 4 Draft

- Created webapp with node.js
- Organized webapp folders and named files

- Created index.hbs, competitions.hbs, and main.hbs files
- Created add_competitions.js, update_competitions.js, and delete_competitions.js files
- started forever
- Used express handlebars

Upgrades Based on Feedback on Step 4 Draft

- We have all the buttons and crud operations working for competitions except update is not working correctly.
- We moved delete from a form below table to a onclick button in table
- We removed the html files from our project folder so it would not render them instead of our hbs files
- We have search now finding and returning the correct data
- We removed the strange LOC string from our app.js file
- We also fixed the td1 variable issues that were in our update_competition.js file

Upgrades to the Project Step 5 Draft

- We have worked long and hard troubleshooting the errors we were making using handlebars and ajax for the first time for both of us. We visited Ta hours twice to get help which was somewhat helpful. We just had to keep working at it. The learning curve for using git also with branches caused some delays for us. But we have now been able to go from one working page to 4 in the last two days. We learned alot about vscode and git also while doing this step.
- We have almost 4 of our 7 pages working with 3 crud operations working for each. 2 of those pages still have one crud operation each left. The other 3 remaining pages have 3 or 4 crud operations each but those pages are very similar to the ones already accomplished.
- We plan on finishing the other 3 pages before Thanksgiving giving us another week to pull everything together.

Feedback By Peer Reviewers - Step 4 Draft

Omotola Anibaba

- Do the implemented CRUD steps function as the team expects (e.g. if the team stated that a CRUD step worked but you found an error, please tell them)?
    - The website looks great, but as another user pointed out, some of the buttons don't work when navigating through the site, though I'm sure they will be fixed as the draft progresses. The 'Competitions' page looks great and is very smooth, but the 'delete' buttons do not work when trying to remove a user. However, this error can be fixed
    -
- Would a user easily be able to use the UI to complete the step? If not or you have suggestions for how the UI can be improved, please elaborate.

    - If the other pages will follow the same flow as the 'Competitions' page, I don't see why not. Everything is easy to follow and has a great flow, whether they are tech-savvy or not!
    -
- What suggestions do you have for the team in any areas where they are blocked or having difficulty? Detailed helpful feedback will receive higher credit.
    - I can see that when I look at each individual html file, they show the corresponding information for each tab, but when I click the link to the actual webpage, I keep getting a 404 error for those same tabs. Maybe there is an error in the routing for each page so that is something to look out for

Review by Aaron Wheaton

Do the implemented CRUD steps function as the team expects (e.g. if the team stated that a CRUD step worked but you found an error, please tell them)?

I'm able to browse the page but like you said none of the buttons seem to work. In particular:

-Delete doesn't seem to do anything

-It appears as though your search works in that it navigates to a URL that is clearly searching for the entered string, but none of the results which you would expect show up, so it's likely not making the proper SQL call to the DB, or the return results are not properly being displayed on the webpage.

-Update did not appear to do anything, and in fact directed me to a 404 result, so it appears as though this disconnect lies within the HTML.

Would a user easily be able to use the UI to complete the step? If not or you have suggestions for how the UI can be improved, please elaborate.

I think you guys have a great UI, and I'm a big fan of the color scheme (dark mode on every single web page user here). The UI is certainly a little busy on the update portion of the CRUD, but I would rather have the ability to access and utilize all the fields I want then have to navigate a bunch of clicks in the name of "minimalism," so I don't really have any recommendations for the UI, good job team!

What suggestions do you have for the team in any areas where they are blocked or having difficulty? Detailed helpful feedback will receive higher credit.

-There's a strange LOC string on the first line of your app.js file that VSCode is identifying as erroneous, I do not know if that was created from a program, introduced somewhere, etc. but that line could possibly be causing some exceptions if it's kicking the app.js out with an error every time it hits that first line of code

-Your update_competition.js file may possible have some issues in the scoping of your td1 variable, that could possibly explain the 404 issues when I attempted to update a value

review by Alexander Rose

Alaxander's review somehow ended up being for someone elses project as first because something was wrong with our link. We fixed it but this review had happened already. So I grayed it out since not relative to us

Do the implemented CRUD steps function as the team expects (e.g. if the team stated that a CRUD step worked but you found an error, please tell them?

It looks like the page button for the Reviews page is working! Not only did the link to the page work and present all the Reviews info but the insert, delete, and update buttons all work too.

Would a user easily be able to use the UI to complete the step? If not or you have suggestions for how the UI can be improved, please elaborate.

A couple nice to haves on the Reviews page would be an indicator showing what the highest possible score is and a larger default size for the review content text box. But overall the page is intuitive and works great!

What suggestions do you have for the team in any areas where they are blocked or having difficulty? Detailed helpful feedback will receive higher credit.

Since the Reviews page is working excellently I would compare the routes for that page to that of the other pages and see if that comparison brings up any differences that could be resulting in all the 404s for the other pages.

Feedback By Peer Reviewers - Step 3 Draft

Reviewer: Renee Sanders

Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.

Yes, the UI does utilize a SELECT for every table and each has sample data.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

There are search functions for each entity but none have the ability to search/filter with a dynamically populated list of properties.

Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.

Yes, the UI does implement an INSERT for every table in the schema with all corresponding attributes listed.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).

Yes, it seems that each INSERT was done correctly with all the corresponding FK attributes. Great job on this! I love the use of dropdowns to choose the various attributes.

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

Yes, every table contains a DELETE operation including one from a M:M relationship.

Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

Yes, every entity has the option for UPDATE. Again, I love the use of the dropdowns!

Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.

Yes, there are a few NULLable relationships.

Do you have any other suggestions for the team to help with their HTML UI?

Great job on this project so far! I like the look of the UI and the use of the description on a home page. The only suggestion I would potentially make would be with the searches for each page. I think there should be some sort of instruction that tells a user what to enter to perform a search. As it is now, it just states to ex: "Search Competitions" with a blank text field. Is the user supposed to enter the name?

Reviewer: Abraham Byun

Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.

Yes, every table has a SELECT query that dynamically displays all the data.

· *Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?*

Yes, every table has a SELECT/search feature.

· *Does the UI implement an INSERT for every table in the schema?* In other words, there should be UI input fields that correspond to each table and attribute in that table.

Yes, every table has a way to insert data for each table in the schema.

· *Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?* In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).

Yes, the event table has multiple foreign IDs used when inserting new data.

· *Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?* In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

Yes, every table has a delete function.

· *Is there at least one UPDATE for any one entity?* In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

Yes, every table has an edit/UPDATE function.

· *Is at least one relationship NULLable?* In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.

Yes, the events table can be created without a competition ID, which can be added later.

·*Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.*

No suggestions. The website and UI is extremely clean and well done!

Reviewer: Jessi Frenzel

I like the look of your UI, especially the colors you've chosen. Also, Cute team name - Apple Jacks!

Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.

Yes. There is a table on each page of the UI and each was constructed with a mysql SELECT query.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

Yes. The competitions entity utilizes a dynamic search.

Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.

Yes. There is INSERT (add) functionality for each table and each page of the UI.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).

Yes.

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

There are at least 5 DELETEs. "DELETE FROM Athlete_Events WHERE ____" (the last DMQ query) does affect an M:M relationship.

Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

Yes, there are several UPDATEs.

Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.

Yes. Athletes can have their team or division become NULL if the team is deleted.

Do you have any other suggestions for the team to help with their HTML UI? For example using AS aliases to replace obscure column names such as fname with First Name.

- p.9 of the PDF - I think you meant to say "intersection table" instead of "transition table".
- I think the font you chose looks nice but it could be easier to read if it were a more basic font like Arial. The same for your PDF.

Good work!

Reviewer: Kyle Fuller

Does the UI utilize a SELECT for every table in the schema? In other words, data from each table in the schema should be displayed on the UI. Note: it is generally not acceptable for just a single query to join all tables and displays them.

Yes, Select is utilized for every table in the UI.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?

Yes, there is at least one search function utilized.

Does the UI implement an INSERT for every table in the schema? In other words, there should be UI input fields that correspond to each table and attribute in that table.

Yes, INSERT is implemented for every table in the UI.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship? In other words if there is a M:M relationship between Orders and Products, INSERTing a new Order (e.g. orderID, customerID, date, total), should also INSERT row(s) in the intersection table, e.g. OrderDetails (orderID, productID, qty, price and line_total).

Yes, Insert works correctly with corresponding FK attributes.

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship? In other words, if an order is deleted from the Orders table, it should also delete the corresponding rows from the OrderDetails table, BUT it should not delete any Products or Customers.

Yes, DELETE function is used with every table including a M:M relationship.

Is there at least one UPDATE for any one entity? In other words, in the case of Products, can productName, listPrice, qtyOnHand, e.g. be updated for a single ProductID record?

Yes, UPDATE is utilized for every entity.

Is at least one relationship NULLable? In other words, there should be at least one optional relationship, e.g. having an Employee might be optional for any Order. Thus it should be feasible to edit an Order and change the value of Employee to be empty.

Yes, there is at least on NULLable relationship.

Do you have any other suggestions for the team to help with their HTML UI?

I have no suggestions.

Feedback By Peer Reviewers - Step 2 Draft

Reviewer: Annika Ewers

- Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?
    - Yes, the schema presents a physical model that follows the database outline and the ER logical diagram exactly.
- Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
    - Almost everything is consistently named. There are a few inconsistencies between the ER and the schema (some things in the schema are snake_case and they are camelCase in the ER). There are both kinds of naming inside of the schema, so it might be best to choose just one. Other than that, the entities are plural and the attributes are singular. The entities are capitalized and the attributes are lowercase.
- Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?
    - Yes, the schema is easy to read. The relationships are clear and the lines are not crossed.
    - (The one thing I would say might be a tiny bit confusing is where the line meets up at the Divisions table but I'm not even sure if there's a way to change that in PHPMyAdmin, and it's not really that big of a deal)
- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?
    - Yes, the intersection table is formed properly with two FKs and facilitate a M:N relationship.
- Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?
    - I do not believe there are any partial or transitive dependencies based on the sample data.
- Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)
    - Yes, the SQL file is syntactically correct.
- In the SQL, are the data types appropriate considering the description of the attribute in the database outline?
    - Yes, the datatypes are appropriate.
    - (The Athletes_Events table isn't listed in the outline, which I'm not sure if it has to be, but I saw the data types in the schema!)
- In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?
    - Yes, primary and foreign keys are correctly defined. It looks like appropriate CASCADE operations are declared.

- In the SQL, are relationship tables present when compared to the ERD/Schema?
    - I believe the relationship tables are present when compared to the ERD/Schema.
- In the SQL, is all example data shown in the PDF INSERTED?
    - Yes, the example data shown in the PDF is in the SQL.

Nice job group 4! :)

Reviewer: Ming-Yang Chang

Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?

Yes, the schema shown here is a great representation of a physical model that follows the database outline and ERD.

Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

For the most part, there is consistency in naming such as entities being plural and capitalized and attributes being singular and lowercase. But as mentioned by another review there is an inconsistency where both snake case and camel case are being used and I do agree only one should be used.

Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?

The schema is easy to read, all the attributes and important keys within the entities are present and labeled. Relationship lines between entities are clear except for the green lines where there seems to be an overlap between two lines.

Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

Yes, the relationship is properly formed with the amount of FKs in affected entities. But the relationships between entities are shown more clearly in ERD compared to the schema. The lines in the schema don't really tell me what kind of relationship those entities have.

Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?

The sample data shown here doesn't seem to have any issue that I am aware of.

Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)

There aren't any issues when applying SQL code to PhPMyAdmin. Tables were created and values or inserted without any issues.

In the SQL, are the data types appropriate considering the description of the attribute in the database outline?

Yes, the data type description in SQL matches the database outline.

In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?

Yes, the primary and foreign keys are correctly defined in SQL in comparison with the schema. On delete cascade are being utilized and appropriately declared.

In the SQL, are relationship tables present when compared to the ERD/Schema?

Yes, when using the designer tab in PhPMyAdmin, relationships between entities matches tables presented in ERD and schema.

In the SQL, is all example data shown in the PDF INSERTED?

Yes, all example data shown in the PDF were inserted in the SQL.

Reviewer: Brooke Ryan

- Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?
    - Yes, the schema follows the database outline and the ERD exactly.
- Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
    - Yes, all the naming appears consistent in terms of the pluralization and capitalization of attributes in the overview, outline, and ERD. Just a styling note, not that it's essential to any functionality, but I would recommend having the same ordering of the attributes between the outline and the ERD and Schema.
- Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?
    - Yes, the schema was easy to read, no relationship lines crossed and I thought coloring every relationship line differently was a good choice by making it easy to follow the relationships when relationship lines were in close proximity to one another.
- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?
    - Yes, the intersection table *Athlete_Events* demonstrated the M:N relationship between Athletes and Events, and contained the two FKs of *eventID* and *athleteID* as well as a composite concatenated PK of *athlete_eventID*.
- Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?
    - The sample data does not suggest any obvious conflicts with normalization.
- Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)
    - Yes, the SQL file looks fine; no issues were encountered when submitting the DDL queries to PhPMyAdmin.

- In the SQL, are the data types appropriate considering the description of the attribute in the database outline?
    - The data types seemed appropriate to track events for what the outline refers to as "hundreds and sometimes thousands" of competitions; this implies there would likely not be an occurrence that would warrant a BIGINT data type for things like PK unique IDs for each entity, as INT would likely encapsulate all instances of the baton-twirling competitions.
- In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?
    - CASCADE operations were declared and the PKs and FKs are correctly defined based on the schema.
- In the SQL, are relationship tables present when compared to the ERD/Schema?
    - Yes, the same relationships defined in the ERD/Schema are implemented in the SQL.
- In the SQL, is all example data shown in the PDF INSERTED?
    - Yes, the outline contains visualizations of the sample data inserted into each of the Project's corresponding entities. One suggestion I had for the Athletes_Events table, since this is a M:N relationship make sure to include data to demonstrate instances where an athlete can attend many events (which you did by showing the Athlete with the AthleteID of 1 attending the EventIDs of 5 and 2), but you should also include some sample data to show where one event featured many athletes (e.g. maybe insert another instance that corresponds to the eventID of 5 and the AthleteID of 2)--to fully demonstrate this M:N relationship, you would need to include sample data that would demonstrate a :M relationship can occur in both directions.

Reviewer: David Harlan

Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?

The Schema seems to match up with the ERD, but I had a bit of trouble figuring out the mandatory vs optional symbol in the schema. All the tables and attributes seem to match though.

Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

Athletes_Events.athlete_eventID has no underscore on the ERD. I couldn't see any other naming discrepancies. Capitalization and plurality all seems fine to me.

Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?

As I mentioned before, not sure how to tell mandatory and optional relations on the schema. Green lines connecting Events and Divisions are connecting, so might want to adjust that. Other than those items it looks good.

Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

Intersection table is Athletes_Events and contains two FK's. All looks fine to me.

Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?

Normalization looks good. All tables are 3NF I believe.

Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)

I had no issues adding DB to PhPMyAdmin. Syntax worked fine.

In the SQL, are the data types appropriate considering the description of the attribute in the database outline?

Datatypes all look good to me and match the SQL.

In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?

PK and FK's are properly defined in the SQL. The ON DELETE operations are declared. There are no ON UPDATE declarations, but I think that's fine since it defaults to "no action".

In the SQL, are relationship tables present when compared to the ERD/Schema?

Relationships properly match in SQL compared to ERD and schema.

In the SQL, is all example data shown in the PDF INSERTED?

Yes, the tables that were inserted into in the SQL were correctly shown in the draft.

Nice job group 4!

Feedback By Peer Reviewers - Step 1 Draft

Reviewer: Ravindu Udugampola

- Does the overview describe what problem is to be solved by a website with DB back end?
    - *Yes, the overview states that the a database driven website with a form interface will be used to record the event entries by hundreds of athletes at Baton Twirling Competition. Although it is not presented as a solution to a problem, a DB backend is ideal for the situation presented here. I would suggest distinguishing what a Competition is and what an Event is in the overview because it was somewhat confusing.*
- Does the overview list specific facts?
    - *Yes, the overview states that there will be hundreds of athletes who will enter one or more events. I would suggest including additional numerical information about the number of events and approximated participants expected for each event.*
- Are at least four entities described and does each one represent a single idea to be stored as a list?

    - *Yes, Location, Competition, Events, and Athletes are some of the entities that each represent a single idea. Some tables like Athlete_Events were not defined in the outline and the objective of it was not clear. Level was a confusing entity because it entails a difficulty level (Advanced, Beginner etc.). I would suggest changing it to a category table because if I understand the logical structure there should only be a limited number of Levels.*
- Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?
    - *The outline of entity does detail the purpose of each except for Athlete_Events. The outline also does list the attributes, datatypes, and constraints between entities.* I found however some Relationships were not correctly identified. For example, the FK teamID was said to be included inside Athletes, but in the ERD it is inside Team.
- Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?
    - *The 1:M relationships are correctly formed: Team to Items (1:M), Coach to Athletes (1:M), Location to Competition (1:M). The M:M relationship of Athlete_Events was not correctly formed as it should include athleteID and eventID as FK. Check athlete_events, I think we have this correctly and putting athlete_events in the outline should fix the confuision.*
- Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
    - *Naming conventions followed in the overview is followed in the entities and attribute creation. The entities are all plural and attributes are all singular. Capitalization is done only for entities and attributes are all lowercase which is the convention recommended in the course.*

Reviewer: Gregory Navasarkian

-Does the overview describe what problem is to be solved by a website with DB back end?

The document specifies how the database will be used and solve the problem of collecting baton twirling competition information. The database will store information about various competitions, events, and athletes.

-Does the overview list specific facts?

The document does not specify the amount of information that is expected to be stored at any given time. The overview specifically lists the tables that are involved in the database. I would suggest adding some specific numbers like expected usage and database size.

-Are at least four entities described and does each one represent a single idea to be stored as a list?

Yes, there are nine entities in total, and each represents a single idea. Each entity is unique and stores information related to various competitions and competitors.

-Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?

Yes, the document outlines connections between entities. The attributes are descriptive and unique to each entity. The document explains how, as an example, each team can only have one coach and so on. The attributes are descriptive and easy to understand.

-Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?

The 1:M relationships are correctly represented in the document as well as the diagram. The document contains two M:M relationships. However, they are not represented in the diagram. The diagram does not contain an M:M relationship. The ERD does present a logical view of the database. There are some small errors in the document that are not present in the ERD. I would recommend altering the diagram to reflect the descriptions of the relationships.Check for errors in ERD and outline, Check M:M relationships

-Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

The naming in the database is unified. All the entities are following the same scheme and all the attributes and keys are in the same pattern. All the keys follow the same naming convention, and the names are all lower/camelCase.

Reviewer: Nathaniel Deshazer

- Does the overview describe what problem is to be solved by a website with DB back end?
    - Yes! Entries for a baton twirling competition.
- Does the overview list specific facts?
    - The attributes for each table and plenty of background information is provided, but rough numbers for how many athletes will attended how many competitions would be beneficial.
- Are at least four entities described and does each one represent a single idea to be stored as a list?
    - Yes, more than enough entities around a single idea.
    - Team seems like an entity you don't need. I would suggest just making team an ID attribute in Athletes and Coaches. You could make it NOT NULL to fulfil the 'each athlete must have a team' requirement. I feel the same way about the Division entity. It might work better as an attribute in Athletes. This way you can just do an inner join between events and athletes with division selected. You could even sort by division and see each athlete by that category and whatever other data you need.
- Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?
    - Yes. Very descriptive.
- Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?
    - I believe all of the relationships are correct. I would suggest adding the Athlete_Events onto your list of tables if you are going to have it in your ERD.
    - Also, must each athlete have a division or would that be optionally 1? Again check Division and add athlete_events to the outline

- Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
    - Yes! All naming requirements were properly met.

Reviewer: Jarod Lokrantz

Does the overview describe what problem is to be solved by a website with DB back end?

Yes, the overview section of the draft lays out the purpose of the database is to support an online entry form for multiple baton twirling competitions by storing records of entries into said competitions. A DB backend is essential for this task for retrieval of the competitors info at a later date when needed.

- Does the overview list specific facts?

Yes, the overview explicitly states that the database will contain one transition table (intersection table?) and that the form will support 100s of athletes.

- Are at least four entities described and does each one represent a single idea to be stored as a list?

Yes, according to the overview, the DB back end will record entity information on competitions, events, teams, athletes, coaches, divisions, levels, and locations. From the lecture, I know that DB design is a squishy topic with multiple solutions, but some of those entities personally sound like they would be better suited to be attributes of other entities.

- Does the outline of entity details describe the purpose of each, list attribute datatypes and constraints and describe relationships between entities?

Yes, the outline details the purpose of each entity and each contains reasonable attributes with reasonable datatypes. As others have pointed out, it does not detail the purpose and attributes of the intersection table needed in their schema, however.

- Are 1:M relationships correctly formulated? Is there at least one M:M relationship? Does the ERD present a logical view of the database?

Yes, the 1:M relationships exist and are reciprocated in the relationship outline of the associated entity detail outline. There is one M:M relationship between events and athletes which makes sense for a M:M relationship. The ERD is mostly reasonable, but the one big stand out issue I could find is that the relationship between events and competitions is optional in both directions, meaning events can exist without competitions and competitions can exist without events, which is somewhat confusing and could use clarification.

- Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

Yes, the naming is consistent and Capitalized in the right places to my knowledge, but the entity names are not plural.

Project Title

Baton Twirling Competition Entry Form Database

Team Name

Apple Jacks

Authors

Amy Mummert

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.

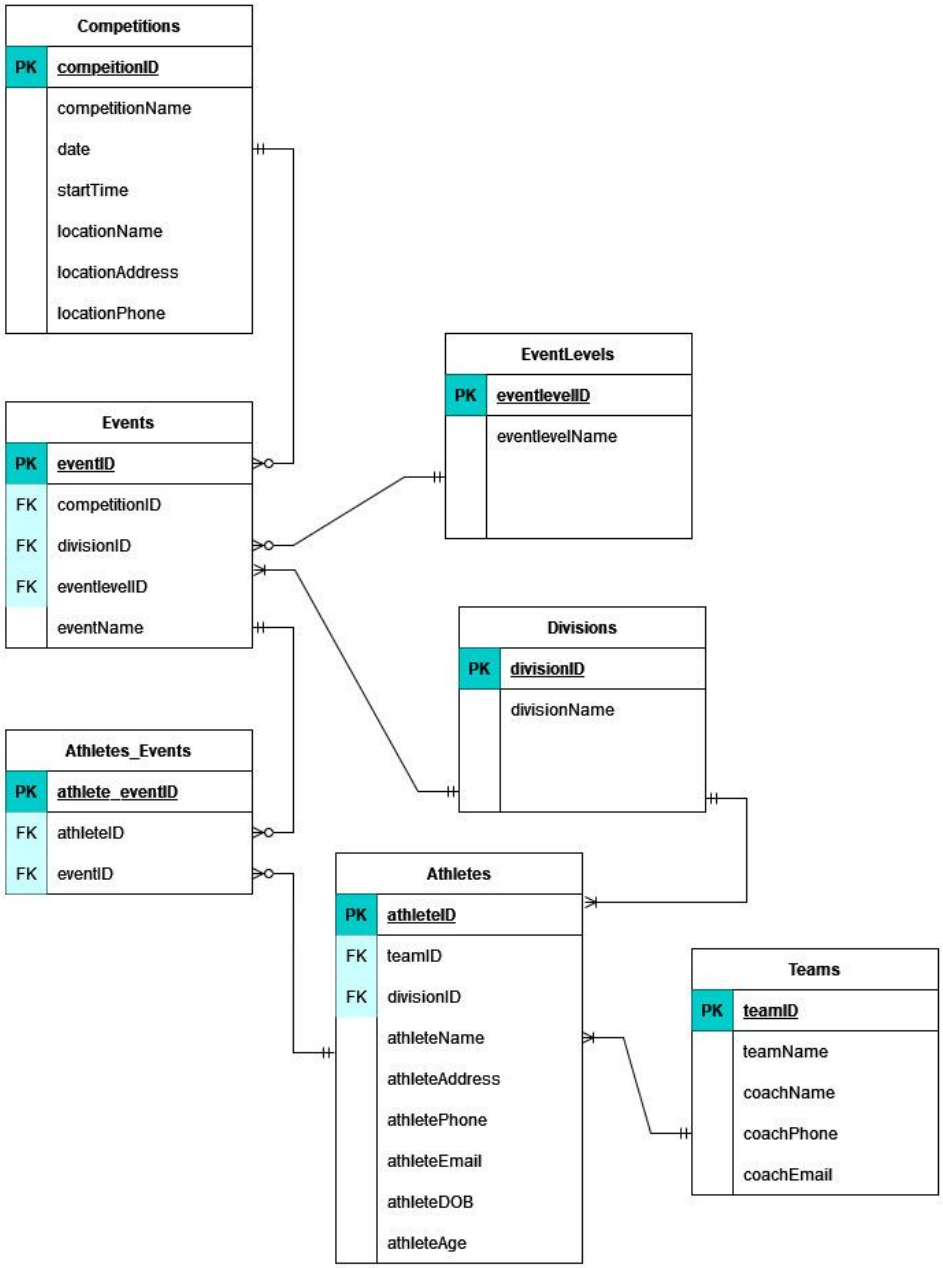
Divisions: Records details for the event and athletes division.

- 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.

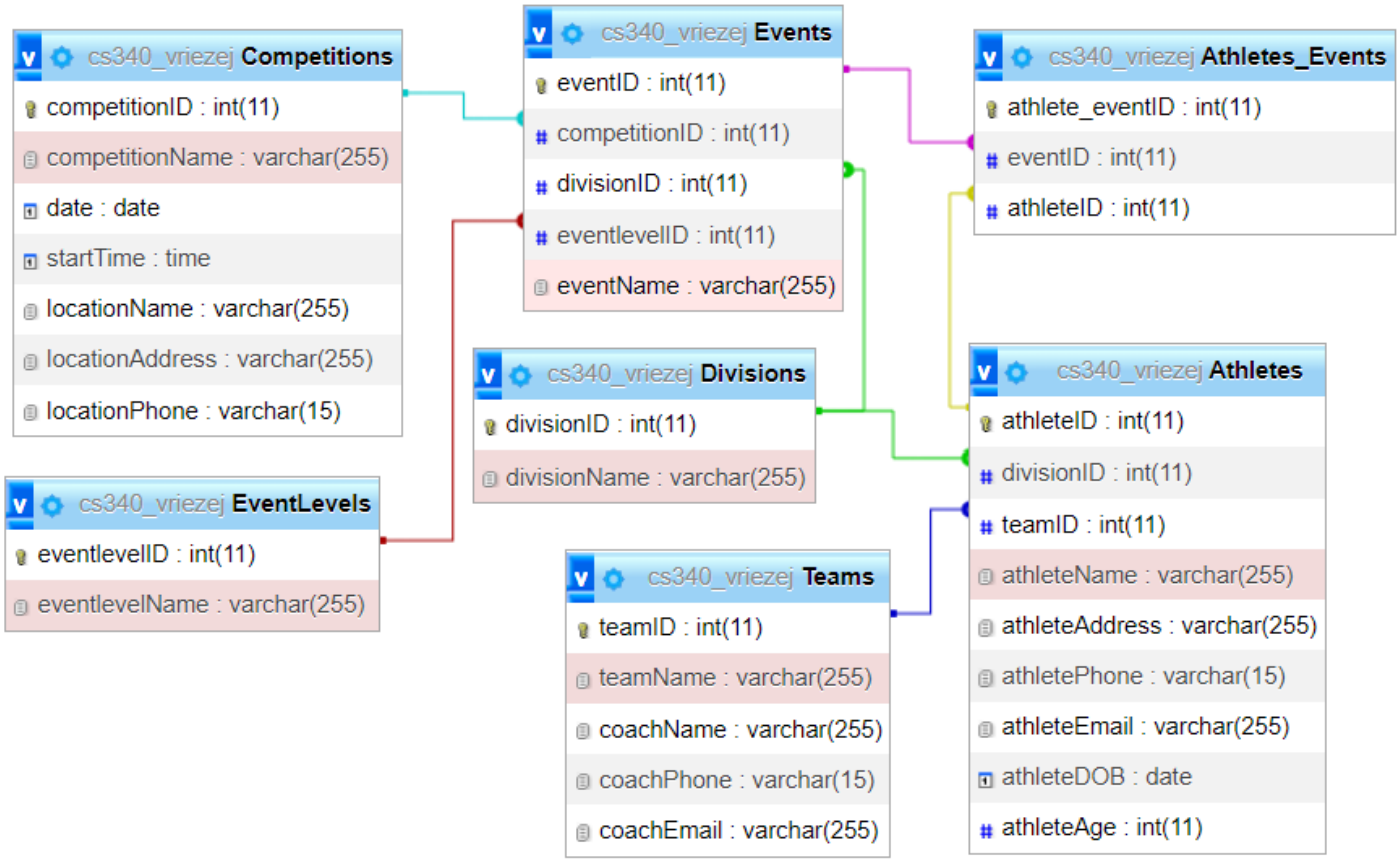
EventLevels: A Category table for event levels. It Records details for each event's level detail in baton twirling.

- 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)



Schema



Example Data

Competitions:

Database: cs340_vriezej, Table: Competitions, Purpose: Dumping data

competitionID	competitionName	date	startTime	locationName	locationAddress
1	Kelly Kadet Baton Extravaganza	2023-03-04	09:00:00	Knova Gym	1000 SE 182nd Ave. Portland, OR 97233
2	Oregon State Baton Championships	2023-04-29	08:30:00	Knova Gym	1000 SE 182nd Ave Portland, OR 97233
3	2023 Western Regional Championships	2023-06-18	09:30:00	Heritage High School Gym	Heritage High School 101 American Ave, Brentwood, CA 94513

Athletes:

Database: cs340_vriezej, Table: Athletes, Purpose: Dumping data

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 62481-04160	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 93855	1-347-476-2792	pede@outlook.edu	2015-07-21	8

Teams:

Database: cs340_vriezej, Table: Teams, Purpose: Dumping data

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:

Database: cs340_vriezej, Table: EventLevels, Purpose: Dumping data

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

Divisions:

Database: cs340_vriezej, Table: Divisions, Purpose: Dumping data

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:

Database: cs340_vriezej, Table: Events, Purpose: Dumping data

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:

Database: cs340_vriezej, Table: Athletes_Events, Purpose: Dumping data

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