Roller Coaster Database Documentation

Group 24: Alex Densmore and Logan S

Project Outline

General Description and Entity Definitions

This project will involve creating a relational database that keeps track of information about roller coasters, amusement parks, amusement park owners, roller coaster manufacturers, and roller coaster features. Definitions of each of these four entities are below, followed by a more detailed outline of entity attributes and relationships.

- Roller Coaster - An amusement park ride that involves a train on a track that is allowed to freely coast at some point during the duration of a ride. A roller coaster is distinct from other types of amusement park rides not included in the database, such as carousels and ferris wheels. A roller coaster will only be added to the database once it has opened to the public.
- Amusement Park - A location designated for entertainment that involves at least one ride (whether or not that ride is a roller coaster) and can include other attractions such as food stands, live shows, and games. An amusement park will not be added to the database until the amusement park has opened to the public.
- Manufacturer - A company that designs roller coasters. Amusement parks contract out to outside manufacturers for the designing and construction of roller coasters. A manufacturer will only be added to the database once it has opened for business.
- Roller Coaster Features: A roller coaster can have various features such as a launch, loops, special trains (such as trains on which passengers lie down or stand up), and going backwards. A feature can be added to the database at any point, regardless of whether or not any coaster with that feature has been built. SImilarly, a coaster can be added without being linked to any special features since it may not have any special features.
- Park Owner: The company (such as Six Flags or Disney), the family, or the individual who owns an amusement park. A park owner can be added to the database once it opens for business (whether or not it currently owns any specific amusement parks).

Entity Attributes

Roller Coaster Attributes (All are required except Amusement Park):

- Primary ID (auto-incremented integer when roller coaster is added)
- Name - string of characters (100 maximum) indicating the name of this particular roller coaster (such as “The Raptor” or “Millenium Force”). A roller coaster must have exactly one name. However, there is the possibility that there are multiple coasters with the same name.
- Amusement Park - unique integer id of amusement park at which the coaster is located (avoids ambiguity of multiple amusement parks with the same name). A coaster can be located at zero or one amusement parks, but an amusement park can have multiple roller coasters. In the edge case that a coaster opened at one amusement park but then was moved to another, its most recent location will be listed.
- Manufacturer: Int indicating the unique ID of the manufacturer who built this roller coaster. A coaster must have exactly one manufacturer. The same manufacturer can have built multiple roller coasters.
- Year Opened - integer indicating the year in which the roller coaster first opened to the public. A roller coaster must have exactly one year opened (in the edge case that it was moved from one amusement park to another, the date it opened at its first location will be listed). There can be multiple coasters that opened in the same year. The year in which it opened must be between 1900 and the current year since the coaster must have already opened to the public.
- Height - integer indicating the height of the roller coaster in feet. Must be between 10 feet tall and 999 feet tall (validated on client side since MySQL does not support CHECK constraint). A roller coaster must have exactly one height (which indicates the highest point reached by the roller coaster), but there can be multiple roller coasters with the same height.
- Max Speed - integer indicating the maximum speed reached by the roller coaster in mph. Must be between 10 and 149 mph (validated on client side since MySQL does not support CHECK constraint). A roller coaster must have exactly one max speed, but there can be multiple roller coasters with the same max speed.
- In Operation (boolean value) - indicates whether the coaster is still in operation or whether this is a historical record of a coaster that has been closed down. A roller coaster must either be in operation or not in operation (but not both). However, multiple roller coasters can be in operation while multiple others can be out of operation.

Amusement Park Attributes (All Are Required):

- Primary ID (auto-incremented integer when amusement park is added)
- Name - string of characters (100 maximum) indicating the name of this particular amusement park. An amusement park must have exactly one name. It is possible for Amusement Parks to have the same name.

- City - string of characters (100 maximum) indicating the name of the city that the amusement park is located in. An amusement park must be located in exactly one city. Multiple amusement parks can be located in the same city.
- State/Province - string of characters (100 maximum) indicating the name of the state or province (or equivalent administrative division of a country) that the amusement park is located in. An amusement park must be located in exactly one state/province, but more than one amusement park can be located in the same state/province.
- Country - string of characters (100 maximum) indicating the name of the country that the amusement park is located in. An amusement park must be located in exactly one country, but more than one amusement park can be located in the same country.
- Owner - integer ID of the Park Owner who owns this park (from the Park Owner table). A park must have exactly one owner, and an owner can own 0 or more parks.

Manufacturer (All Are Required):

- Primary ID (auto-incremented integer)
- Name - string of 100 characters maximum indicating the name of the company. A company must have exactly one name, but multiple companies can have the same name.
- City - string of 100 characters maximum indicating the city in which the manufacturer's primary headquarters are located. A manufacturer must have exactly one city of primary headquarters, but a city can have multiple manufacturers.
- State / Province - string of 100 characters maximum indicating the state / province in which the manufacturer's primary headquarters are located. A manufacturer must have exactly one state / province of primary headquarters, but a state / province can have multiple manufacturers.
- Country - string of 100 characters maximum indicating the country in which the manufacturer's primary headquarters are located. A manufacturer must have exactly one country of primary headquarters, but a country can have multiple manufacturers.

Roller Coaster Features (All Are Required):

- Primary ID (auto-incremented integer when a feature is added)
- Feature Name - string of characters (100 maximum) indicating a possible feature that a roller coaster might include.

Park Owner (All Are Required):

- Primary ID (auto-incremented integer)
- Name - string of 100 characters maximum indicating the name of the company, family, or individual who owns the park. An owner must have exactly one name, but multiple owners can have the same name.
- City - string of 100 characters maximum indicating the city in which the owner's primary headquarters are located. An owner must have exactly one city of primary headquarters, but a city can have multiple owners' headquarters.

- State / Province - string of 100 characters maximum indicating the state / province in which the owner's primary headquarters are located. An owner must have exactly one state / province of primary headquarters, but a state / province can have multiple owners' headquarters.
- Country - string of 100 characters maximum indicating the country in which the owner's primary headquarters are located. An owner must have exactly one country of primary headquarters, but a country can have multiple owners' headquarters.

Relationships between Entities:

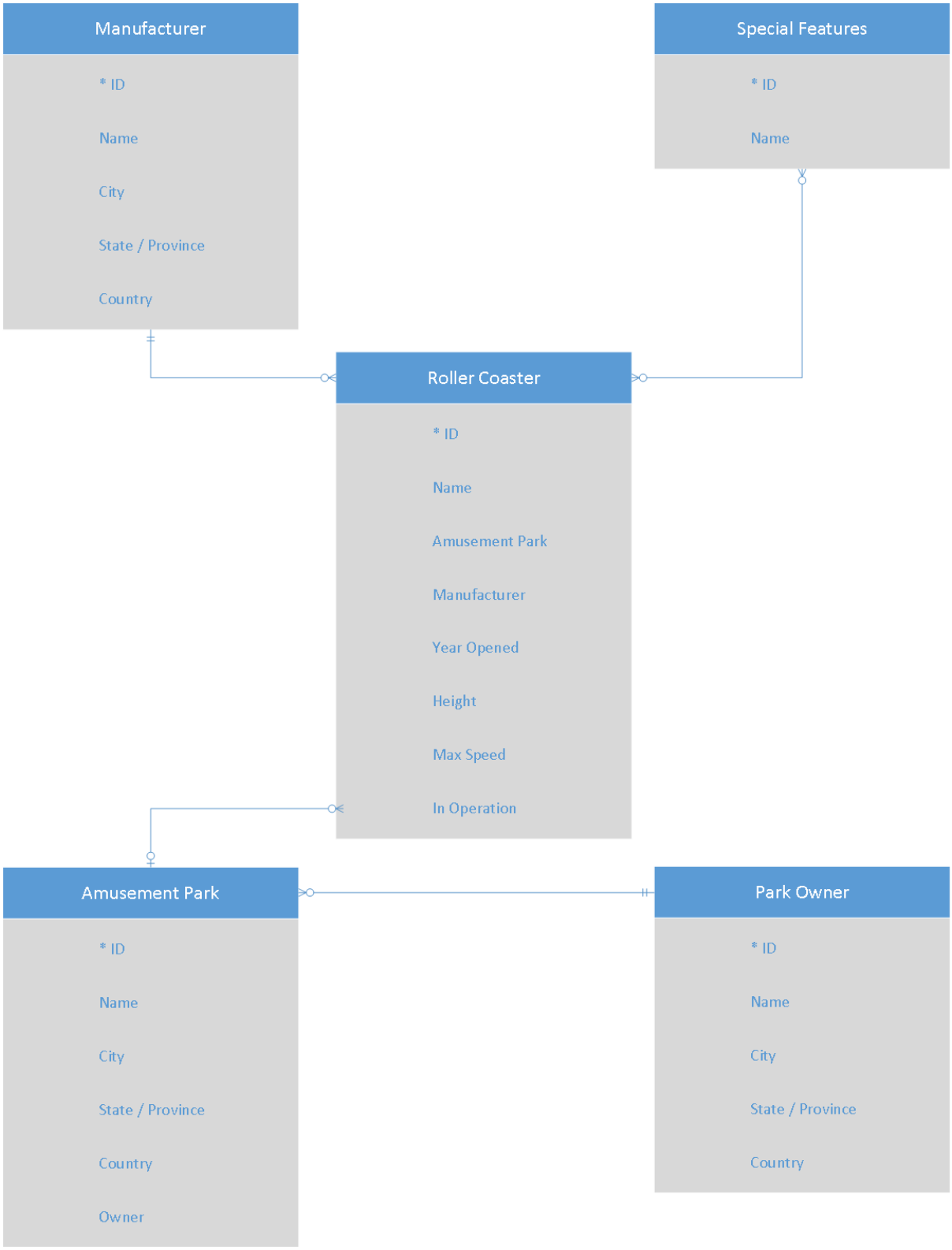
Roller Coaster and Amusement Park: A one-to-many relationship since a roller coaster can belong to 0 or 1 amusement parks but a park can have 0 or more roller coasters. Since this is a one-to-many relationship, the park at which a roller coaster is located is included as an attribute in the Roller Coasters entity table.

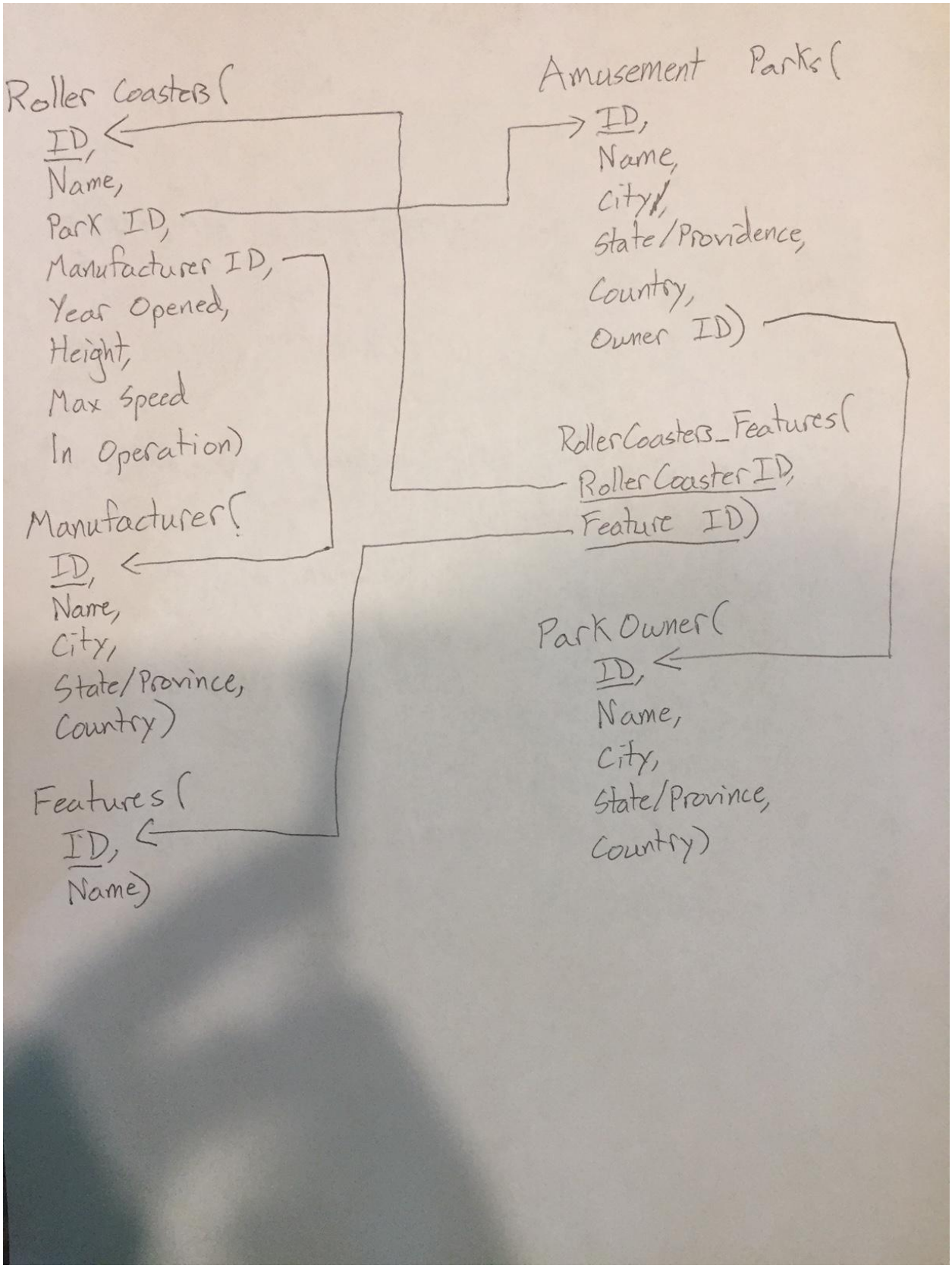
Roller Coaster and Manufacturer: A one-to-many relationship since a coaster must be built by exactly one manufacturer but a manufacturer can have built 0 or more coasters. Since this is a one-to-many relationship, "Manufacturer" is included as an attribute of the Roller Coaster entity table.

Amusement Park and Park Owner: A one-to-many relationship since an amusement park must belong to exactly one owner but an owner can own 0 or more amusement parks. Since this is a one-to-many relationship, the owner of an amusement park is included as an attribute in the Amusement Park entity table.

Roller Coaster and Special Features: A many-to-many relationship since a roller coaster can have 0 or more special features and a special feature can be associated with 0 or more roller coasters. This many-to-many relationship will be indicated in a separate table with coaster int id and special feature int id from corresponding entity tables.

ERD and Schema


Manufacturer
* ID
Name
City
State / Province
Country
Special Features
* ID
Name
Roller Coaster
* ID
Name
Amusement Park
Manufacturer
Year Opened
Height
Max Speed
In Operation
Amusement Park
* ID
Name
City
State / Province
Country
Owner
Park Owner
* ID
Name
City
State / Province
Country


Roller Coasters(
ID,
Name,
Park ID,
Manufacturer ID,
Year Opened,
Height,
Max Speed
In Operation)
Amusement Parks(
ID,
Name,
City,
State/Providence,
Country,
Owner ID)
RollerCoasters_Features(
RollerCoasterID,
Feature ID)
Manufacturer(
ID,
Name,
City,
State/Province,
Country)
Park Owner(
ID,
Name,
City,
State/Province,
Country)
Features(
ID,
Name)

Changes based on Feedback on Previous Drafts

Part 6 Peer Review Feedback and Changes

Review from Eric B
All delete functionality was implemented correctly. I was able to delete both rollercoasters and parks without any problems.

Changes based on Eric's Review:
Eric did not suggest any changes.

Review from Tucker S
Good job adding the DELETE functionality

Coasters: The delete functionality worked as expected. The entries were removed when the delete button was clicked.

Special Features: I was able to delete features on a coaster. This worked as expected.

As the Project guide states -"You need to include one DELETE and one UPDATE function in your website, for any one of the entities" meaning you already meet the minimum number of DELETE functions on the "Coasters" page.

Changes based on Tucker's Review
Tucker did not suggest any changes.

Feedback from Robert E
Hey!
Everything is looking great! All the required functions are working from what I could see. The only thing I could come up with to suggest is outside the project parameters, but what if a company or manufacturer went under or moved. Maybe a delete and edit functions for those pages would make sense to me, but again, is not required by this project.
Great job!

Changes based on Robert's Review:

We are not going to be making any changes based off Robert's review. It would be nice to implement the delete and edit functions for all of the pages, but the project does not require this functionality so we will not be making these changes.

Part 5 Peer Feedback and Changes

Feedback from Eric B

Was able to successfully update all fields for a given coaster.

Changes based on Eric's Review:

- Eric did not suggest any changes.

Feedback from Tucker S

Good job adding the UPDATE functionality

Coasters: The update functionality worked as expected. I like the notification that alerts you when you update an entry.

Special Features: I was able to add additional features on a coaster. This worked as expected.

As the Project guide states -"You need to include one DELETE and one UPDATE function in your website, for any one of the entities" meaning you already meet the minimum number of UPDATE functions on the "Coasters" page.

Changes based on Tucker's Review:

- Tucker did not suggest any changes.

Feedback from Robert E

Hey,

Everything looks great on the update functions. The validation was working perfectly for me for speed and height, however, there was none for the year.

You may think about adding that validation so that people can't put invalid years in.

Overall everything looks great and working great! Good job!

Changes based on Robert's Review:

- We have already added input validation for the year. At the time of Robert's review the input validation for the year had not been put in place yet, but now it is.

Feedback from Cody

Hello!

The site is looking good so far. The only thing I would think about is possibly adding update capabilities to the Special Features, Parks, Park Owner and Manufacturers pages. Its not a huge deal as it is easy to delete and re-add the correct data, its just a quality of life suggestion.

Good job!

Changes based on Cody's Review:

- We are not going to make any changes based off Cody's review. The update functionality for all of the entities that Cody is suggesting would be nice to have, but it is not required for this project so we do not have any plans to make those changes at this time.

Changes We've Made on Our Own:

- We edited client-side javascript so that, once a user submits a form and the data has been successfully added to the database, the page refreshes so that the form clears.
- We added functionality to hide table for parks page if there are not any parks in the table, as well as coaster_features table on coaster_features page if there are no features. Also added functionality to have page not allow new features submission if coaster has all features.

Part 4 Peer Feedback and Changes

Feedback from Kyle T

Hey!

Your site is really nice and clean. I tried adding stuff on every page I could and it looks like it is all working as intended.

Just a small detail; I would put the search bar on the coaster page somewhere at the top instead of below the data table. No big deal, just I need to give some feedback!

Again, everything is clean and good and works right.

Good work!

Changes based on Kyle's Review:

- We have moved the search bar to the top so that users can more easily filter for a specific coaster.

Feedback from Cody

Your site is really coming along nicely!

I couldn't find too much to critique at the moment.. I would recommend maybe a better layout of the page. Something like moving the tables and photo's side by side rather than scrolling down to the input section.

Other than that it looks really good!

Cody

Changes based on Cody's Review:

- We are not going to make any changes based off Cody's layout suggestion. The layout changes that he recommends are really a personal preference. We are more concerned with the site and database meeting the requirements of the project.

Feedback from Eric

Group 24 Review

Does the INSERT form actually work for entities and relationships, as required in the Specs?

Was able to insert into all the entities.

Does INSERTing rows in the "M entity" of the 1-to-M relationship rows affect the INSERTing of rows in the "1 entity"?

Was able to insert into one to many relationship.

Does INSERTing rows in the "M entity" of the M-to-M relationship rows affect the INSERTing of rows in the other "M entity"?

Was able to insert into many to many relationship.

Can data be inserted for all entities?

Yes.

Anything else that you think is important for the CREATE functionalities?

The only real issue, which is mostly cosmetic, is the forms not clearing after submitting the form and adding to the database.

READ functionalities

Are rows being listed for all entities?

Was unable to view, special features relationship. Also was unable to view the amusement park to roller coaster relationship.

Are rows being listed for all relationships, as described in the Specs?

Yes all are being liste, minus what was mentioned above.

Changes based on Eric's Review:

- We will edit client-side javascript so that, once a user submits a form and the data has been successfully added to the database, the page refreshes so that the form clears. We will implement this change for our next draft since we did not have the chance to implement it for this draft.

Feedback from Caleb

As was stated in the other review, it looks like all of the boxes that needed to be checked for this one have been checked, and your SELECT and INSERT functionality has been added. The only suggestion i would have would be maybe to have your add functionality be hidden behind a modal with a button, just for readability. As it is right now, having to scroll down to see the add functionality, if you don't know its supposed to be there or are looking for it, might make it hard to find. Either way, good work.

Changes based on Caleb's Review:

- We are keeping the insert form at the bottom so that users read through the table first and verify that a coaster they want to add is not already in the table.

Part 3 Final Version Feedback and Changes

We did not receive any feedback on Part 3 final version requiring changes.

We decided to get rid of the separate many-to-many relationship page for coasters and their special features due to the possible confusion of multiple coasters having the same name. We

want to ensure that users view all basic identifying information about a coaster before viewing or updating that coaster's special features. Therefore, the only way to view and update a coaster's special features will be to click the "Special Features" button next to that coaster's information on the "Coasters" page. The special features page for a specific coaster will list that coaster's basic information from the "Coasters" page and let the user view, delete, or add special features for the coaster.

Part 3 Peer Review Feedback and Changes

Review from Eric B:

Data Manipulation Queries:

- Queries looked good, Did have delete queries in the file, but not used on the webpage.

Date Definition Queries:

- SQL file imported correctly and as expected.
- Foreign keys and relationships match.
- In rcdb_coaster year_opened may be better if use a Date or Year type instead of int

HTML Pages

- All the pages looked good. Was unable to find a form/page for updating and deleting to the database.

Changes Based on Eric's Feedback:

- We decided to leave the 'year_opened' attribute as an int. After researching the 'Year' datatype, it is very limiting and only allows a date range of 1901-2155. While realistically a year stored in the database would most likely be in that range, I think it is fine to just use an int to allow more flexibility (for example, maybe it is a historic record of an experimental coaster in the late 1800s). The "date" datatype would not work well because it requires a month and day, which are not important in this case.
- We are not implementing a separate delete form since a delete button will just be next to deletable entries and will take care of the deletion. We are also not implementing our update form that the edit button leads to quite yet because that form will be the same as the insert form for a given entity except with the current data pre-filled, something that we will not be able to implement until connecting this to the database / server. We are adding an explanation to the html pages clarifying how to view / edit / delete special features of specific coasters since those are in a separate many-to-many relationship table in the database and will therefore use their own forms.

Review from Robert E:

Everything was easy to read and follow.
The only thing I was unsure about was the edit features part of the coaster page. Is this the update function? And if so, does it require you to select the feature from a drop down? Or does it create a new feature and add it to the coaster?

I guess this would be cleared up in a functioning sight, but it was unclear to me at the moment.

One design to think about is including the id in the tables, that way the data seems to progress instead of being in random order. It may make it easier to read and reference later when using the website.

Overall nice job.

Changes Based on Robert's Feedback:

- For the edit features concerns that Robert had, we explained a few updates in the second bullet point of the changes based off Eric's feedback. The "Edit" button on the coaster's page will allow the user to update all of the coaster's information EXCEPT the roller coaster's features. The roller coaster's features will need to be updated by clicking the "Features" button from a roller coaster in the coasters page.
- We decided not to add the id of the entries to the tables that are displayed to the web-page user, because it is not information that the user needs to see. It is only information that the developer, server, and database need to know. (See list of changes we've decided to make on our own for how we have decided to order this information instead.)

Review from Cody P:

I'm very impressed with the website progress so far! I can tell you have put a lot of time into it and it shows.

I only have a couple small details to mention..

Robert mentioned the features page a little bit in his post. I would recommend expanding the table slightly. Rather than just a name column, you could have a title, and a description which would hold more detailed information.

I understand why you display the "Features of specific coasters" on its own page with the way you have laid out the tables directly on the page. However, on a typical website, would that information not be combined into one "Coasters" page? For example, on the one page it

would show the Maverick coaster with its specific attributes (park, manufacturer, etc.) and then a list of its Many-to-Many attributes (Launching, Looping, etc.). This would be so that users can see all info on one page rather than moving pages to view the data.

Those were my only ideas.. Looks great!

Changes Based on Cody's Feedback:

- We chose not to have a description go with the title of a feature because it would violate the 3rd normal form. We decided in a previous draft step to remove a description attribute from the features table entirely to solve this problem.
- We chose not to add the features of each roller coaster on the coaster page. If we did this, it could display a lot of extraneous information. We also needed a page to update the coaster-feature relationship, so it makes sense to display this information on that page instead.

Changes We've Decided to Make on our Own

- Change name of "Coaster" page to something like "All Coasters and their Basic Features;" change name of "List of All Features" page to "List of All Special Features;" change name of "Features of Specific Coasters" to "All Coasters and their Special Features."
- Add drop-down to "All Coasters and their Features" to select a specific feature and see the names of all coasters that have that feature.
- We agreed that we wanted to sort the information that is displayed in the tables in alphabetical order. Since the current hard-coded sample data is just a placeholder, we are not sorting the sample data at this point, but we will use ORDER BY clauses when we connect the front-end HTML to the backend.

Part 2 Final Version Feedback and Changes

We have not received feedback from the graders for Part 2 Final Version requiring any changes.

Changes We've Made On Our Own:

In order to allow a foreign key for one of the one-to-many relationships to be set to NULL, we have updated our outline and ER diagram so that a roller coaster can have 0 or 1 amusement parks (instead of being required to have exactly 1 amusement park). We have also updated the ER diagram and outline to allow an amusement park owner to own 0 or more amusement

parks (instead of 1 or more amusement parks) so that the owner can still exist in the database even if all amusement parks they own are deleted.

We also added clarification that a coaster's height and max speed will be validated to make sure they are within valid range on the client side upon learning that MySQL does not support the CHECK constraint.

Part 2 Peer Review Feedback and Changes

Feedback from Caleb S (Review Group 25):

ERD:

The only thing I noticed is that the Roller Coaster to Amusement park relationship is shown to be 1 to 1 on the ERD, but should be a 1 to many. It is described as a 1 to many relationship earlier in the outline, so this is most likely a simple typo from your diagram program. Also, I'm not sure if your ER diagram should contain foreign keys as attributes, but it looks like you included the amusement park and manufacturer ids in it.

Schema:

Schema looks good as far as I can tell, good work.

Changes Based on Caleb's Review:

We will not be making any changes based on Caleb's review. After double-checking our ER diagram, it already does correctly show Amusement Park to Roller Coaster as a one-to-many relationship. Furthermore, foreign keys are included in entities since they are an attribute of that entity.

Feedback from Miles M (Review Group 19):

I like the roller coaster database! The entities all make sense connected to each other, and it covers all the requirements as far as I can tell. One thing I would suggest is changing your (Roller Coaster - Amusement Park) and (Roller Coaster - Manufacturer) relationships from one-to-many to zero-to-many to match your descriptions in the ER diagram.

Changes Based on Mile's Review:

We will not be making any changes based on Miles' review. From our understanding, there is not a zero-to-many relationship that is available in the notation that we are using for this class. The only relationships available are:

- One-to-one
- One-to-many
- many-to-many

Feedback from Robert E (Review Group 19):

I really like the ERD and Schema. Everything looks good, so I am just going to throw out a few points to think about.

In your description, it says that all Roller Coaster attributes are required, but what if a manufacturer has a roller coaster with special attributes, but its not installed yet. Could the amusement park attribute be set to NULL at that point? Or what about a roller coaster that is installed but not open yet, could year open be NULL then?

Also, the park to special features relationship. Since features are dependent on roller coasters, I am not sure if you need the park-features relationship. If you need to know what features are in the park, you could look at the roller coasters and then looking at coaster-features. But if there can be no roller coasters in the park, could the park have features then too? I am just not sure if that relationship makes since since features are dependents on the roller coasters.

Changes Based on Robert's Review:

We decided to explicitly state in the entity definition that a roller coaster must be installed in the amusement park and be opened to the public before it can be added to the database. We realized that the amusement park to features relationship is not necessary, because you can look up all of the coasters in a park and then find the features that are associated with those coasters. We decided to remove the park and feature relationship. In order to make sure that we had the four required relationships we added a park owner entity and a one-to-many relationship between park owners and amusement parks.

Changes We've Made On Our Own:

We did not receive any feedback from the graders for our outline.

To make sure the database design meets 3rd normal form, we have removed the “description” attribute from the “Special Features” entity so that the ID is the primary key and the name of the attribute is itself a minimal key dependent on the primary key, but there are no transitive dependencies since there are only 2 attributes total.

We also have specified for the Amusement Park, Manufacturer, Special Features, and Park Owner tables that all attributes are required and cannot be null.

In addition, just like we added the clarification of when roller coasters can be added to the database based on Robert’s review, we added clarifications of when rows for all other entities can be added to the database in the “General Description and Entity Definitions” portion of the outline.

Part 1 Final Version Feedback and Changes

We have not received feedback from the graders for Part 1 requiring any changes.