Hudson Dean
Luke Nam

Part A) Updated Outline, ERD, and Schema

Chemical Inventory Database

Project URL: http://flip2.engr.oregonstate.edu:4351/
and: http://flip3.engr.oregonstate.edu:8666/

Project Outline:

We will be making a chemical inventory database. This database will store information on chemicals (also known as compounds) that are synthesized as potential therapeutic candidates for drug discovery. The database will encompass all the information regarding chemical properties, which container it is stored in, and where that container is located within a storage rack. There can be different types of containers that can hold different amounts of chemical and different types of racks to store the containers. The database will simulate real-world application of chemical management in biotech company.

Database Outline, in Words:
The entities in my database are:

- **Chemical** - The chemical entity represents all the different chemicals that are synthesized to be tested in biological experiments as a potential therapeutic candidate. It has the following attributes:
    - **Chemical ID (int)**
        - This number is automatically assigned to each chemical when they are recorded in the database. It is an auto-incrementing number which will be the primary key.
    - **Chemical Name (varchar)**
        - The chemical name is a varchar with a maximum of 1000 characters. It cannot be blank and there is no default. The chemical name must be unique.
    - **Chemical formula (varchar)**
        - The chemical formula represents the chemical structure in a varchar with a maximum of 100 characters. It cannot be blank and there is no default.
    - **Molecular weight (float)**
        - The molecular weight represents the mass of the chemical. It will have a float value with three digits after the decimal point (ex. 32.001) It cannot be blank and there is no default.

- **Container Type** - The container type entity represents the different types of containers that will be used to store the chemical entity. It has the following attributes:

  - **Container Type ID (int)**
    - This number is automatically assigned to each container type when it is recorded in the database. It is an auto-incrementing number which will be the primary key.
  - **Container Type Name (varchar)**
    - The container type name is a varchar with a max of 100 characters that describes the container type. Ex. "1 ML MATRIX VIAL". This value cannot be blank and there is no default.
  - **Container Max Volume (uL) (int)**
    - The container max volume describes the max capacity of the container. It will be an integer that represents the max volume in uL. This value cannot be blank and there is no default.

- **Container** - The container entity describes the contents inside the container. The container entity can store multiple chemicals or be empty. It has the following attributes. The containers will always hold chemicals in liquid form.
  - **Container Barcode (varchar)**
    - All containers come pre-barcoded with a unique numerical varchar. Ex. "C463". The container barcode will be the primary key. This cannot be blank and there is no default.
  - **Container Type (int)**
    - The container type is the ID from the Container Type entity which the container belongs to. This cannot be blank and there is no default.
  - **Concentration_uM (float)**
    - The concentration is a float value. It can be blank if there is no chemical inside the container. There is no default value.
  - **Amount_uL (float)**
    - The amount represents the volume of chemical inside the container. It is a float value that cannot exceed the max volume of the container type. It can be blank if there is no chemical inside the container and there is no default value.

- **Rack Type** - The rack type entity represents the different types of racks that will be used to store the various containers. The rack features a grid where the vials can be stored inside. It has the following attributes:
  - **Rack Type ID (int)**
    - This number is automatically assigned to each rack type when it is recorded in the database. It is an auto-incrementing number which will be the primary key.
  - **Rack Type Name (varchar)**
    - The rack type name is a varchar that describes the type of rack. The varchar has a max of 100 characters. EX. "1 ML MATRIX VIAL RACK". It cannot be left blank and there is no default. The rack type name must be unique.

- **Max Capacity (int)**
  - The max capacity is an integer which represents the max number of containers that can fit inside the rack. It cannot be blank and there is no default.
- **Allowed labware type (int)**
  - Each rack type can only store one container type. The allowed labware type is the ID from the Container Type entity. It cannot be blank and there is no default.

- **Rack** - The rack entity describes a physical rack where the containers can be stored in. Based on the Rack Type, the rack will be able to store a max number of containers inside of it. It has the following attributes:
  - **Rack Barcode (varchar)**
    - All racks come pre-barcoded with a unique varchar. This is the primary key. This cannot be blank and there is no default.
  - **Rack Type (int)**
    - The rack type is the ID from the Rack Type entity which the rack belongs to. This cannot be blank and there is no default.
  - **Cells occupied (int)**
    - The cells occupied attribute is an integer that represents the number of containers that occupy the cells inside the rack. This value cannot exceed the max capacity of the rack type. This value is an integer between 0 and 96. This value cannot be blank and the default value is equal to the max capacity.

Relationships:

Containers have multiple chemicals - A container can have one or more chemicals that can be stored inside it. The same chemical can be present in multiple containers. The container and chemical entities have a *many-to-many* relationship. Has table of chemical_in_container.

There is one Container Type to a Rack Type - a Container Type can only fit inside one Rack Type and a Rack Type can only be associated with one Container Type. The Container Type and the Rack Type entities have a *one-to-one* relationship

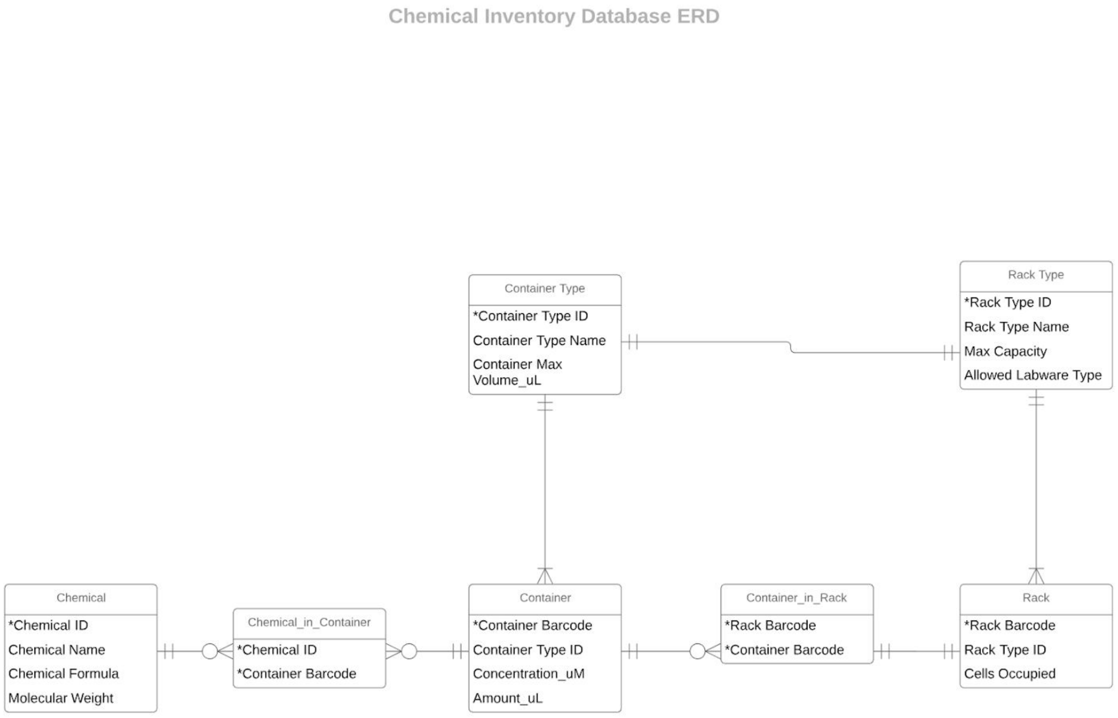
Each Container has one Container Type - a Container has only one Container Type that it specifies to but multiple Containers can have the same type. The Container and Container Type entities show a *one-to-many* relationship.

Each Rack has one Rack Type - a Rack has only one Rack Type that it specifies to but multiple Racks can have the same type. The Rack and Rack Type entities show a *one-to-many* relationship.

Hudson Dean
Luke Nam

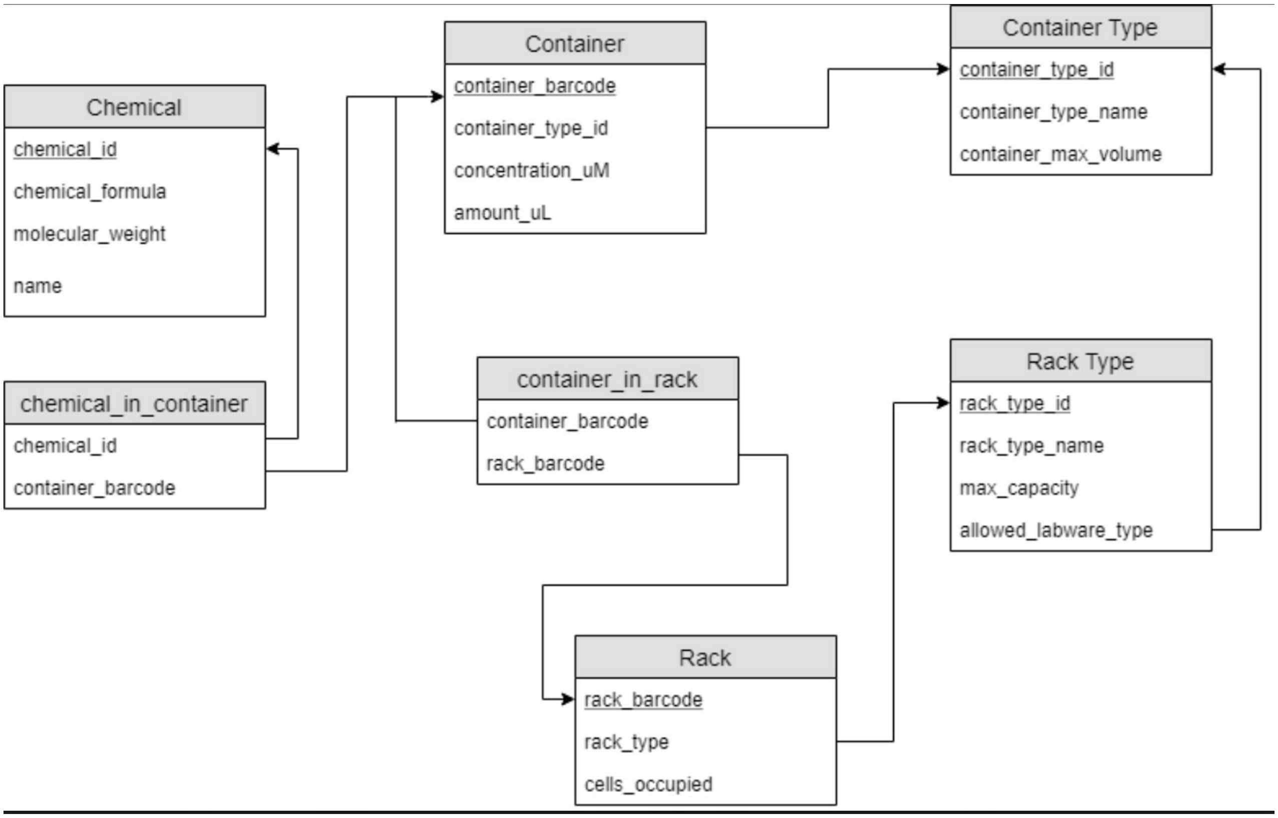
A Container can only be found on one Rack - A specific Container entity can be found on one and only one rack. A Rack can hold multiple containers. The Rack to Container relationship is a *one-to-many*. Has table of container_in_rack.

Entity Relationship Diagram Updated:



Hudson Dean
Luke Nam

Schema Updated:



Part B) Fixes based on Feedback

The feedback from Project Step 1 was to include data types for our attributes in each entity. We have now changed the outline to include the data types for each attribute. Some data types were specified in the description of the attributes but not all. Now, data types can be found next to attribute names in parentheses.

We did not receive any new feedback for Step 3 (2/28/19).

Step 6 Feedback by the peer reviewer:

Review From Kyle Schuetz

The site looks great! I was able to successfully update and delete some records in your site. It appears that you have correctly implemented the update functionality with a form that is pre-populated with all of the individual values of the record to be updated. This makes the update process flow nicely and allows the user to determine what they would like to update and what they would like to remain the same.

Additionally, I noticed that you have abstracted away the need for a user to know individual IDs in the edit container form. This is a great practice and makes your app much more user friendly!

The only feedback I have for your app right now that isn't a congratulations on your solid work up to now is in reference to the Add Rack form. It appears that the drop down menu for Rack Type Name is not populating so you can't actually input a Rack Type Name into the database from the form, it returns a database error each time you try to submit any data.

Over this was a very solid effort!

Review From Kyle Schuetz

Your website looks good and the links all seem to be working and such. I like that you have the top navigation bar, however, when I set my browser to half of the screen to write this, I do notice that the nav bar does not quite come outright. You might want to look into this as it covers some of the page. The pictures feel like they should redirect you to the corresponding tab when the picture is clicked on which you might want to add functionality for. I like how the site has been organized in the chemicals tab. It is nice that we can add chemicals, search the inventory and move chemicals.

Delete & update:

The update and delete buttons look very cool. The chemical is easy to edit with one press of a button that corresponds to a row. I love how you did this. When I added some updates to some of the chemicals you had it worked very well and smoothly. I also deleted the magnesium citrate row which worked very well removing it from the list and updating the page immediately. Overall it looks great and I was surprised that these features existed for most other pages on the nav bar.

It looks great and worked great based on my tests! Keep up the good work!

Review From Yan Liong Tan

Draft 6 Delete.

Hi Hudson,

I am assuming since you didn't post step 6 you mean to merge step 5 with 6.

Here are some recommendations:

1) When you press delete on the chemical page. You give a pop up that just shows the word delete. Maybe the you could be more specific in saying that something was deleted

2) When I try to create another a chemical with the same name, and formula the web page presents me an sql line "

{"code":"ER_SUBQUERY_NO_1_ROW","errno":1242,"sqlMessage":"Subquery returns more than 1 row","sqlState":"21000","index":0,"sql":"INSERT INTO chemical_in_container(chemical_id, container_barcode) VALUES ((SELECT chemical.chemical_id FROM chemical WHERE chemical.name = 'test'),'00009')"}"

In the ddq for chemical you could have a UNIQUE constraint that constrains name, molecular weight and chemical formula to stop me from adding the same chemical

3) When I move a chemical from one container to another, the container is still remembered in the previous container. So now I have 2 chemicals that are in 2 different containers. Meaning the delete operation did not occur when you move a container

Best Regards,

Liong

Review From Matt Dienhart

Hudson,

Most of my comments are about the update functionality. I just have two comments for the draft 6 submittal to do with the delete functionality:

1. Deleting a chemical seems to work, but there is a pop-up message that is shown after the delete button is clicked. Maybe you are trying to get this pop-up to have a confirm/cancel option to prevent accidental data loss, and if so, that's a cool idea. However, in its current state it is just an OK button, so it's just providing an extra step in the delete process for no reason.
2. Deleting container types seems to work if there are no references to that container type from other entities, but once you create a container and assign it the new container type, it cannot be deleted anymore. If this is the intended functionality, it would be good to show an error message explaining the reason it can't be deleted. Otherwise, if this is not the intended functionality, then you should double-check the ON DELETE behavior of your container type entities.

I hope this helps, good luck finishing the project!

-Matt

Step 6 Actions based on the feedback:

Feedback That Was Implemented:

1. Removed chemical deleted alert
2. Containers can not be deleted as of now
3. Chemical name is now a unique attribute
4. Can now change and update chemicals in containers on the container page after searching
5. Fixed broken link of pictures on the homepage that was not navigating to the corresponding page
6. Add rack form was implemented

Feedback That Was Not Implemented:

1. Did not implement a move chemical from one container to another. The only functionality is being able to add a chemical to an existing container. One chemical can be in multiple containers.

Step 6 Upgrades to the Draft Version

1. Updated DDQ file
2. rack location attribute was removed from container_in_chemical table
3. updated ERD/Schema to reflect rack location removal
4. Made datalist instead of select to allow user to choose from the list or enter their own input. A drop down with pre-populated data gets unwieldy if the table has many rows so it is convenient that the user can enter their own input.
5. Added search chemical inventory function
6. Removed chemical formula from search

Step 5 Feedback by the peer reviewer:

Review From Tam Bowen

Hi Yan,

First, I think you should have another post for draft 6, based on Professor's comment on Piazza a few days ago (see the comment below)

Hudson Dean
Luke Nam

"Since the functionality implementation is vastly different for both UDPATE and DELETE, we have two different assignments and hence please make separate posts. It's okay if you post the same content in both.

It also helps the other group members in focusing on one aspect while reviewing, helping you and learning from your project."

The Update/Delete for the chemical page works great. Some minor things that I suggest:

1- please check your container type's html. For some reason, it display an extra "L" at the end of all Container Type name.

2- maybe you could have a hyperlink for the image on the home page

3-Also, there is one thing I am not sure how your website could do some error check: I could create a container barcode for a 4 ml vial and place it within 2ml vial type rack.

Anyways, that is all I have. Your website looks great

Review From Yan Liong Tan

Draft 5

Hi Hudson,

Here are some recommendations I have for draft 5:

1) Rack doesn't allow updating of the information. As a user I should be allowed to rename my rack. If I cannot rename a rack, I would have to delete the rack and create a new rack and reassign or items in the original rack to this new rack.

2) The same addition of an update function for container and rack type applies.

3) Update container failed for the container with a blank container name, what would happen is that the site shows the container page again instead of the edit page of the container, but with the addition of a container header that suddenly appears from the top of the page. Your dqm for updating the container seems correct and the ddq has the correct not null for setting the barcode of the container when creating the container table ; I think it is the in the code, you may not have set the values properly or mapped the values to the wrong attribute in the table to cause this bug

Best Regards,

Liong

Review From Matt Dienhart

Hudson Dean
Luke Nam

Hudson,

Since you say the chemicals page is the furthest along, that is where I focused my review. Here are my Draft 5 comments:

1. When I try to create or update a chemical, the barcode and rack location are always blank, but strangely the concentration, amount, and rack barcodes are always set to "6000", "180", and "R123" respectively. Your SELECT query in your DMQ looks OK, but your INSERT query for a new chemical is missing a container barcode value, so that partially explains it. For the other attributes, maybe check to see if the web page is not rendering correctly?
2. When I try to update a chemical, I am taken to a page that has all of the attribute values of that chemical already populated in an input form, except for "container barcode", which is pre-populated as "blank" no matter what it was before. Maybe this is related to problem #1?
3. On the chemical update page, clicking the "back" button appears to have the exact same result as clicking the "update" button. The "back" button should cancel the edit. Make sure these buttons are assigned to different routes in your application & HTML.

I hope this helps. Good luck!

-Matt

Step 5 Actions based on the feedback:

Feedback That Was Implemented:

1. Fixed html adding extra L to container type name
2. Can now update the rack type and container type names
3. Changed the code queries to better reflect DMQ queries
4. Changed container barcode field in the chemical update page to not be set to empty
5. Fixed back button updating fields in update pages

Feedback That Was Not Implemented:

1. Did not include hyperlink image to home page, home nav bar tab works fine for our purposes
2. Error check is not a priority as we are focused on finishing implementing update and delete features
3. We do not want the user to alter the container or rack barcode as they are primary keys. If want to change a either, container/rack have to be deleted

Hudson Dean
Luke Nam

Step 5 Upgrades to the Draft Version

1. Upgrades are listed in Step 6 Upgrades

Step 4 Feedback by the peer reviewer:

Review From Leif Tsang

Hi Luke,

Your project looks great! I took a look at your website and it definitely looks well done. I tried to put some stuff into your database and it was very smooth. Things updated right away which I liked very much. Your output is well formatted and looks well organized. After adding some stuff to your cells I realized that your count was off and while you might be planning on fixing that later, I thought I'd let you know. I had trouble adding a container with a random int as the barcode. I had some sort of child process error and it could be because I need to use a specific container barcode. Overall, your site looks very good and I could totally see the site being practically used.

Leif

Step 4 Actions based on the feedback:

Feedback That Was Implemented:

1. No feedback was implemented because there was no negative feedback.

Step 4 Upgrades to the Draft Version

1. None

Step 3 Feedback by the peer reviewer:

Review From Kyle Schuetz:

When I review your ERD I see that it is possible for you to get information about all of the chemicals that are housed on each Rack. Are you planning to have a query that displays all of this information? I could definitely see a use case for an end user to want to be able to view this information as they could then use this information to deduce patterns in your chemical storage. This might be a great feature to add to your app (unless I have completely missed the feature in my analysis). Looking at your DML queries it appears that you have queried down to the container level but have not gone to the level of each individual chemical (perhaps nesting -- Search container and view contents inside the query --Display contents inside rack and how many cells are occupied would get you the desired result).

Overall, I think your group has a really solid start on this project. All of your data definition queries appear to be creating the correct tables, with the correct relationships. Additionally, all of your data manipulation queries appear to be returning the test data with acceptable performance. As far as your HTML front end is concerned, I find it to be extremely intuitive. I am able to quickly grasp the purpose for each of your individual pages and I am easily able to understand the utility of your delivered solution. In general, I think this is an extremely successful implementation of the requirement for Step 3 of the project. Given the overall quality of your solution provided to this point, I look forward to the continued progression of your project throughout the quarter. Great work!

Review from Hyunwook Shin
Hey Luke,

First off I'm impressed with how much work you and your partner have put into the project so far. I'm not familiar with chemicals and their properties, but seems to lend itself well to being organized into a database. With that said I will start your feedback with the website.

Website:

Since the feedback gets repetitive with so many pages that are similar in scope linked to the index, I will do a general summary to save time. I thought the index page was very laid out with a great summary of what the links were and the description of the functions about each link. I also though having the inventory of the relevant data points with each link was really helpful when it came to testing the different queries. Very user friendly index page! As for the linked pages that housed the different functions, they made sense and lined up with the queries you had posted in your DML file. Again, the UI for the website links were well thought out and I was able to navigate my way through.

Data definition file:

The DML file is well organized and I was able to match the various sections of the file to the implemented tables when I imported into mariaDB. It all seems to match the ERD and schema and I don't see any flaws with the DML file.

Data manipulation file:

I tested a few of these out since there are so many and the ones I tested (add new container, delete container, update container and view all registered container types) worked and worked as you described in your schema and ERD so this looks like it is on track as well.

Overall very well done, I think you've added a lot of useful queries for the user and the design of your project deserves praise as well.

Review from Leif Tsang
Hi Luke,

I dont see anything I can add that the others have not already covered, but everything looks great. I love your table and the topic is perfect. Something that I could definitely see being practical in a chemistry lab. Everything is labeled very well and clearly. Keep up the good work for sure it is looking great. As everyone else also said, your .sql files look good and implemented correctly.

Review by Matt Dienhart

Hudson,

I think your project looks good!

Here are my comments:

Some of the input fields in your HTML forms have up and down arrows that allow you to change the numerical value in increments of 1, but I'm not sure this is useful, especially for things like molecular weight which usually have several decimal places. You can also get negative values by pressing the down arrow, and you probably want to prevent negative inputs wherever possible as part of your input validation.

Your foreign key constraints don't have any defined behavior for when the referenced elements are updated or deleted. You will probably want to add this to your DDQ, otherwise it will be possible to have (for example) a rack holding a container that no longer exists in the database.

You may want to indicate in your HTML forms somehow what the required fields are. For example, if the user wants to add a container, per your DDQ they must provide a barcode and a type, but they can leave the concentration and the amount blank.

Hope this helps! Good luck!

-Matt

<u>Review by Rasheed El Kassed:</u>

Hi,

Everything's look pretty great so far. I've only got minor issues to add since Matthew covered the more major issues.

Inconsistencies between the schema and the actually database. I know this is minor, but you might want to use the schema as a reference later on so it would be nice to have them both be equal.

chemical_name in the schema vs chemical_name in the database.

container_max_volume_uL in the schema vs container_max_volume

rack_type_id (under Rack) in the schema vs rack_type

With regards to the input fields that are type number, another suggestion would be to change the increments to 0.01 instead of just outright removing them. That's your call though

I have nothing left to add. I hope the feedback was useful!

-Rasheed

Step 3 Actions based on the feedback:

Feedback That Was Implemented:

1. Added a query to show the chemical inventory that is inside a container (Kyle Schuetz)
2. Added a query to display the rack contents. This will show which container is in which location inside the rack. (Kyle Schuetz
3. Added foreign key constraints on delete (Matt Dienhart)
4. Inconsistencies between schema and actual database (Rasheed El Kassed)
5. Change input field increments when using arrow keys and allowing user to input negative numbers (Matt and Rasheed). (HTML was updated before that changed these issues)

Feedback That Was Not Implemented:

1. Everything was implemented from Luke Nam’s feedback.

2. Everything was implemented from Hudson Dean’s feedback.

Step 3 Upgrades to the Draft version:

1. Upgraded data manipulation queries to reflect the added functionality of the front end.
2. Added chemical inventory search to display the inventory of a searched chemical and which rack it is located in
3. Added a move chemical form that will move a chemical id into a new container
4. Added a form that will allow the user to enter a rack barcode to display which containers and where the containers are located inside the rack.
5. Added another view that shows the number of cells that are occupied inside the rack
6. Added pictures and a navigation to allow the user to easily get to different pages no matter what page the user is in.
7. Added a script.js file that allows the user to click on a picture in the home page and navigate to the correct page
8. Added styles.css page to make the page look more professional
9. Updated Schema to better reflect the names of tables in database implementation

Step 2 Feedback by the peer reviewer:

Review From Matt Dienhart:

Hudson,

This is an interesting idea for a database. I think you have done a good job of defining your entities and relationships in a way that will provide the functionality that you described in the project outline. However, I think there are some issues with how you defined the attributes for your entities.

Here are my comments:

You are using "string" as a data type for some of the entity attributes. "String" is not a SQL data type, it is a category of SQL data types.
Don't confuse attributes and relationships. Some of the entities you defined, such as Container and Rack, have attributes that are just references to other other entities, such

as "Container Type" and "Rack Type". For example, a Rack's "Rack Type" attribute is just a reference to the ID number of a Rack Type entity. I understand that "Rack Type" will appear in the Schema table for Rack, but I don't think it is correct to call it an attribute of Rack; it is a relationship between Rack and Rack Type.
My understanding of the Rack entity is that each rack has multiple cells where Containers can be stored. Therefore, I am confused by the "Cell Number" and "Cell Container" attributes of Rack. What is the "Cell Number" of a Rack that contains multiple cells? If you want to be able to describe the contents of every cell in a particular Rack, shouldn't you make a new entity called Cells with a one-to-many relationship to Rack? (i.e. one rack contains many cells)
I think your ER and Schema diagrams look correct for the outline you provided, but you may need to make some small changes if you decide to implement the comments I made above.

I hope this was helpful. Please let me know if you have any questions.

-Matt

Review from Tam Bowen:
Hi Hudson,

The outline of your Chemical Inventory Database is very clear and match with the ER model. However, I have a few recommendation for the scheme:

1- Since the relationship between Chemical and Container is many-to-many relationship, having "Chemical ID" as an attribute inside the Container table will not work. "Container" table will only allow one record for each unique "Container ID", so each container can only have one chemical inside, which conflicts with your original design. To make this work, you could make both "Container ID" and "Chemical ID" as the composite key, or you could remove "Chemical ID" attribute in the "Container" table and make another table to track the "Container" and "Chemical" relationship. The second option is the better choice in my opinion.

2- I agree with Matt that the "Cell" should have its own entity. From your outline, "Cell_Number represents the individual cell inside the rack grid", so "Cell_Number" is not a character of the Rack entity, hence not an attribute.

3- (small design change) It is nice to have 2 more attribute for the "Rack" table: 1- to track number of cells are available. 2- to track number of cells are occupied

4- (small design change) Instead of using Chemical ID as the primary key for your "Chemical" table, you could use the existing database called "CAS Registry Number" (known as Chemical Abstract Service Registry Number: Unique numerical identifier assigned by American Chemical Society to every chemical substance described in the open scientific literature). I also recommend to add "Chemical Name" as an attribute, for easy identification of chemicals. However, keep in mind that one chemical could have multiple names (e.g. acetone is also known as 2-propanone, propanone, or dimethyl ketone).

5- Here is a side question: Are you assuming that those chemicals that mix into the same container will not react with one another?

I hope this helps and good luck with your project.

Hudson's Response to Matt & Tam:
Thank you both for your feedback! It was very helpful. I agree, there are some changes that need to be made for some of the attributes (that they need to be changed to relation tables) and some of the entities will need to be changed.

Thank you Matt and Tam for bringing this to my attention!

Review from Yan Liong Tan:
Hudson,

I like your idea for this database, a chemicals storage system is a genuinely useful place to implement a database. The entities and relationships are properly defined that should achieve the goals you have set out in the project outline. However, there is room for improvement.

You have written that chemical is the most important entity in the database however it has very few attributes compared to container and most of the relationships are linked to the container relation. Chemical should at least have included the name of the chemical so that user’s can easily search for the chemical name if they don’t know the chemical formula in the database.

Concentration Unit has two types “uM” and “mM”, you should just have one type and use the type with the smallest unit; Once you build the webpage for the project, client-side javascript can allow the user to change the concentration to uM and mM by converting the smallest type to the largest type. That way concentration will always be the same type and you can increase and decrease concentration and compare concentrations between any container.

Similarly Amount Unit has two types “uL” and “mL”, you should just have one type and use the type with the smallest unit; Once you build the webpage for the project, client-side javascript can allow the user to change the amount to uL and mL by converting the smallest type to the largest type. This way amount will always be the same type and you can increase or decrease amount and compare amount between any container.

The cell container attribute is something I am not too clear about. If it is a copy of the Container’s barcode and we have already linked the container to the rack, is this attribute really necessary information? Couldn’t it be just queried through the container entity linked to the rack?

The Entity Relationship Digram should use id instead of “table name ID”. For example, change Chemical ID to id. This will make the transition the schema and comparison with the schema for other readers more easier .

I wish you well in the project

Hudson’s Response to Yan:

Thank you for your feedback Liong. I found it really helpful. I agree that chemical shouldn't be worded like that, the design has changed in that it is no longer the case that it is most important. Cell attribute is another change that needs to be changed since it does not make sense in its current state. I was not aware of that implementation for the unit type, that is something I will have to look into.

Hudson Dean
Luke Nam

Thank you for bringing this to my attention. Your feedback was very insightful!

Review from Hyunwook Shin

Hi Luke,

Not sure how you and your partner decided to chose your topic but it definitely works well for the purposes of this project. I've broken the review down below in separate sections so you can easily see my feedback.

Database outline in words:

1) All entities are described thoroughly and I can understand the different entities without difficulty.

2) The attributes seem to make sense for each entity and are defined with data types which are appropriate.

3) 5 entities and 5 relationships are described which satisfies the number of requirements for entities and relationships needed for step 2 of this project.

ERD:

1) Crow's notation makes sense between each entity and is consistent with the database outline in words.

2) There are "key" labels next to each attribute of each entity which is confusing. Not sure if you mean to use each attribute as a key creating a giant composite key? If this is the case I would advise creating an auto incrementing ID number for each entity so that your primary keys aren't all complicated composite primary keys.

Schema:

1) The relationship containers have multiple chemicals needs a relationship table. I believe two entity tables that share a many to many relationship that have many rows on each others tables need to have a new relationship table created with both primary keys sent to the new relationship table as a foreign key. So I would suggest making a chemical-container table.

2) All other relationships are one to many or one to one relationships and do not require additional tables.

Hudson Dean
Luke Nam

Overall the project has all the components needed for the final version. I would suggest looking at the schema again and representing the many to many relationship.

Luke’s Response:

I appreciate the feedback about the containers having multiple chemicals needing a relationship table. I was just talking about this with my partner and it is something I'm going to implement into my project outline. This will allow us to make the relationship between multiple chemicals being in the same container. Thanks!

Review from Hao Liu

Hi Luke,

My apologies for posting the review for another group under this thread earlier. I have removed that reply. Here are my comments on your outline, ER diagram, and schema.

1. Throughout the document, it is a bit unclear to me what the Container entity's relationship is to the Chemical entity. Is it 1) a container CURRENTLY stores one chemical, and a chemical CAN be present in zero or more containers (one to many relationship), or 2) a container CAN store zero or more chemicals, and a chemical CAN be present in zero or more containers (many to many relationship)? Your ER diagram indicates that it is a many-to-many relationship, but your outline's introductory paragraph seems to indicate a one-to-many relationship ("...which container it is stored in...").

2. In your outline and your ER diagram, the relationship between Container Type and Rack Type is one-to-one. This means that a rack can only be used to store only one kind of container. It seems a little restrictive. What if someone runs a smaller lab and needs to put different types of containers on the same rack?

3. For containers and racks, how is a barcode different from an ID? If a barcode is unique, why is it necessary to need the ID?

4. For the Rack entity, my understanding is that you are trying to put an array of integers in the Cell Number attribute and Cell Container attribute, so that you can index which container is located in which cell. However, this violates the principle that says attributes in our data model must be single-valued (P57 in the text book).

Hudson Dean
Luke Nam

5. Your ER diagram does not indicate which attributes are the primary keys.

6. On your schema, the Rack entity's cell_container attribute points to the barcode attribute, which is not a primary key of the Container type. This violates the referential integrity constraint, which states that every non-null foreign key value must match an existing primary key value (P99 on the text book).

Luke’s Response:

I appreciate the feedback. Here are my answers to your questions.

2) The relationship is many to many because there can be zero or more chemicals inside one container. The same chemical can be inside zero to many different containers. I will fix the wording in the introductory paragraph so that it is less confusing.

3) I decided on a chemical inventory management system that could be used with automation. These types of containers and racks are restrictive in a one-to-one relationship because you want specific types of labware to be stored inside specific types of racks. I am going to use two different types of containers. The first is going to be a container that can hold 1 mL and the other container that is going to hold 4 mL. The volume differences lead to an increased size for the larger container. In automation, labware follows a standard size format called an SBS footprint. This means that all labware has the same footprint. This allows manufacturers to make robotic equipment that is compatible with all labware.

4) I was initially going to use one table for containers with chemicals inside of them. If there was a container that was going to have two chemicals then it would have two lines with duplicate barcodes. This is why I felt the need to use another ID. I am going to implement another relationship table between containers and chemicals.

5) I plan on having it be a single value. If a rack has 96 cells inside a container then it will have 96 rows in the table.

6) Thanks for catching that. I'll add that in.

7) I didn't know that. I'll go ahead and read page 99 and add that to my outline.

Thanks again for the feedback!

Hudson Dean
Luke Nam

Step 2 Actions based on the feedback:

Feedback That Was Implemented:

1. Removed the saying “Chemical is the most important entity” from outline. (Yan’s suggestion)

2. Changed string data type to varchar data type. (Matt’s suggestion)

3. Address Cell number attributes (Matt’s suggestion).
   - We changed cell attributes to make a one-to-many relation called container_in_rack. This table contains rack_barcode and contianer_barcode; Can be blank and there is no default.

4. Table for the relationship between container and chemical (Tam’s suggestion).
   - We added a table called chemical_in_container to handle this many-to-many relationship.

5. Add name attribute to chemical entity (Yan & Tam suggestion).
   - Adding the name attribute made sense for the purpose of this database.

6. Removed the container id and used container barcode because it is a unique identifier (Hao Liu)

7. Limited user to only work in units of uM for concentration and uL for this project.

8. Added table to build relationship between container and rack. This describes which containers are inside the rack specified by the barcode. The container barcode is tracked based on where it is in the rack location. This was made to simplify the database due to the confusion among most reviewers about the purpose of some of the attributes inside the previous Rack table. (Everyone) ● This goes hand in hand with feedback #3.

9. Removed “key” labels next to ERD which was confusing for the reviewer. (Hyunwook Shin)

10. Added a signifier to indicate which attributes are the primary keys in the ERD

    (Hao Liu).

Feedback That Was Not Implemented:

1. Tam’s suggestion of “3- (small design change) It is nice to have 2 more attribute for the "Rack" table: 1- to track number of cells are available. 2- to track number of cells are occupied”
   - We decided not to add these attributes since we will incorporate the Cell table (one-to-many relation between rack and container). In this table we will keep track of how many cells are occupied in a given rack.

2. Yan’s suggestion to removing Container Type entity and transfer attributes to container.
   - We decided that we will keep Container Type entity since it serves our purpose well and multiple containers can have the same type.

3. Tam’s suggestion on swapping Chemical ID for CAS number or adding CAS as a separate attribute.
   - We decided that an auto incrementing ID for chemical will be easier to use for our purposes in this class rather than use a official CAS number. (Interesting idea!)

4. Matt’s and Tam’s suggestion to change specific ID names in each attribute to just “id”.
   - We’d like to keep the ID/ Barcode names the way they are.

5. Hao Liu’s suggestion to allow more container types to rack types.
   - We would like to keep this relationship one-to-one to restrict one container type to a rack type.

Step 2 Upgrades to the Draft version:

Upgraded and changed ER Diagram
Upgraded and changed Schema
Upgraded and changed Outline to reflect changes in ER diagram and Schema