CS 340 Project Step III Draft for My Kidney Nutrition Tracker Application

Project Group 70-Charles Cal and Deanna Denny

URL: http://flip1.engr.oregonstate.edu:61257/index.html

(have to be connected to VPN)

Fixes Based on Feedback:

Listed below is the feedback we received from the peer reviewers, and what changes we made from their feedback. We also provided below changes we made as well that we decided upon as a group.

Step I Peer Review Feedback:

Peer Reviewer #1:Cassandra Hedrick:

First I just wanted to say that this is an amazing idea that really seems to not only have a real world application but could really help people.

- Does the overview describe what problem is to be solved by a website with DB back end?
    - Yes this overview laid out exactly what the current problem is and how it is affecting a large group of people. Lack of access to kidney disease nutrition and overabundance of misinformation thus causing life threatening impacts to thousands of people.
- Does the overview list specific facts?
    - Yes the overview specifically talks about the 37 million people that are affected by kidney disease and that there are approximately 165,000 applications that seem to be helpful for the restrictive diet that is necessary with the disease but most if not all are not helpful for patients.
- Are at least four entities described and does each one represent a single idea to be stored as a list?
    - Yes there are Patients, Food, Lab_results, Dialsis_forms and more.
- Does the outline of entity details describe the purpose of each, list attribute data types and constraints and describe relationships between entities? Does the outline clearly indicate which entities (tables) will be implemented and which team member is primarily assigned to the associated page(s)?
    - Yes, such as the Food section has a M:M for as it is explained there will be many foods listed and it will correspond to many patients.
    - I didn't see anything listed on specific tasks for each team member.
- 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 looks to be properly formulated.

    - The M:M is for Food and Patient relationship
    - The ERD does look like a logical view from my understanding of the project.
- Is there consistency in a) naming between overview and entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
    - Yes. For example Food is labeled differently than food_id which makes a clear indication with capitalization and the naming between the attributes that they are different. I did not see anything in plural that would give concern on naming.

I think this is a great start to the project. Thank you for sharing it.

Peer Reviewer #2: Clifford Bielinski:

Good job to both of you. Really interesting topic and well thought out project.

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

You do a great job in showing both the scope of the problem at large (kidney disease) and the specific niche that your product is aiming to address among all the health-based apps out there. Perhaps include some metrics on anticipated number of patients that will be using the app when you first roll out.

- Does the overview list specific facts?

Yes - they do a good job describing the purpose of the application and why the entities they're tracking (lab values, dialysis regimen, food content) are important as it relates to kidney disease.

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

There are four entities plus an intersection table. Each one clearly represents a unique concept.

- Does the outline of entity details describe the purpose of each, list attribute data types and constraints and describe relationships between entities? Does the outline clearly indicate which entities (tables) will be implemented and which team member is primarily assigned to the associated page(s)?

I think this is one of the strongest parts in the project outline. The entity descriptions are really great and I have a good sense of what each Entity is about and how it relates to the others. Every attribute has a data type and constraints where appropriate. One thing I question is having separate attributes for values and units of measurement. Perhaps

it's stored as a consistent unit of measurement (e.g. Na as mmol/L) and then patients can input it in a variety of predefined units of measurements at the application level and it's just converted and stored at the database level using consistent notation. Probably would make it easier to analyze the data.

- Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

Yes they have both 1:M and M:M relationships that are properly formulated. They also include an intersection table for the M:M relationship between Foods and Patients. The 1:M seems correct: patients have one form of dialysis at a time and each patient has many lab results.

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

The naming is both consistent and clear. Entities are plural and capitalized. Attributes are lower case and singular with underscores separating words. The attribute names are informative but not overly verbose.

Overall, a really great job!

Peer Reviewer #3: Mark Mendez:

Cool project, guys! I've actually been the person helping a dialysis patient in my family keep track of their diet, so I know the problem this solves. In fact, if you have a feature roadmap that extends beyond this class, I suggest you consider supporting other diet restrictions, such as prohibited foods for transplant patients on immunosuppressants, as a later feature.

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

Yes. The website will help dialysis patients track their diet, which is a difficult task.

- Does the overview list specific facts?

Yes. The overview numbers the potential users as well as some market-related quantities.

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

Yes. Patients, Food, Lab_Results, Dialysis_Forms, and Patient_Food each represent a single idea that makes sense as a list.

- Does the outline of entity details describe the purpose of each, list attribute data types and constraints, and describe relationships between entities? Does the outline clearly indicate which entities (tables) will be implemented and which team member is primarily assigned to the associated page(s)?

Yes. The descriptions are especially helpful, because they introduce readers to the concepts used in the table which are not common knowledge. As many have noticed, the assignment page did not prompt to note likelihood of implementation or triage details.

- Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

Yes. 1:M and M:M relationships reflect their intended purpose. I found a minor detail that could be adjusted for consistency between the ERD and word outline. Patients do have a M:M relationship with Food, but this relationship is implemented with food_id and patient_id as foreign keys inside Patients_Food.

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

The Patients_Food table has attributes food_id and patient_id, which I expected to be prefixed as Food_food_id and Patients_patient_id to match the convention used for other attributes. Otherwise, everything is consistent.

Peer Reviewer #4: Joshua Nees:
Interesting project, Deanna and Charles! You guys have done a great job showing the need for building the data and how it might be used to improve patient outcomes.

Here are my notes:

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

The description gives a great overview of why the database is needed and who it will serve.

Does the overview list specific facts?

Yes, specific facts are listed which give a good indication of the scale of the inputs expected. It may be worth adding additional info about how frequently a patient expects lab results.

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

Yes.

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

Yes.

Does the outline clearly indicate which entities (tables) will be implemented and which team member is primarily assigned to the associated page(s)?

The entities are clearly indicated. No information about team member assignments is given.

Are 1:M relationships correctly formulated? Is there at least one M:M relationship?

Yes, the relationships look correct and there is an M:M relationship between Patients and Food

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 appears consistent.

Step II Peer Review Feedback:
Peer Reviewer #1: Kyle Gifford

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

Yes, a physical model was included and it seems to follow the outline and ER diagram.

- 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?

I wasn't able to detect any inconsistencies and all capitalizations seemed correct.

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

I found it very clear and easy to read.

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

All intersection tables are formed properly.

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

The sample data do not suggest any non-normalized issues.

- 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!)

The SQL file is syntactically correct.

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

In the SQL all data types seem to be appropriate.

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

Proper CASCADE operations were declared in the intersection table "Patient_Foods" and primary and foreign keys were correctly defined.

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

The relationship tables are consistent with the ERD/Schema.

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

All data is inserted.

Conclusion:

Great job on your project so far! It is a really admirable idea. You implemented all of the changes as recommended by previous reviewers and have a really great draft that I found no issues with.

Step III Peer Review Feedback:

Peer Reviewer #1: Nabrissa Khan

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 in the schema. Data from each table within the schema is displayed in the UI. However, a small change is that the names of the tables do not match the capitalization that is currently on the DDL query file. On the statements for Create Table, they all appear to be lower cased and that differs from what is currently shown on the PDF document for the Schema model. For example, it is showing as “Patients” on the Schema model on the document, but `patients` on the Data Definition queries document itself on the separate SQL file.

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

There are more than one SELECT statements that utilize a search/filter with a dynamically populated list of properties. The Foods entity/table does have a search bar that is featured on that part of the web page. I anticipate that this will eventually lead to a dynamically populated list of properties for this table. These are featured well on the DML file, but it will depend on how we will see it on the UI. There may need to be updates to queries to have the user be able to have the option to select by food_id rather than have an overwhelming list of options to pick from.

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.**

The UI implements an INSERT for every table in the schema. The DML looks great and there are INSERT statements for every table within the schema. These are indicated for all five entities of Lab_Results, Patients, Patients_Food, Dialysis_Forms, and Foods. I saw this was indicated for Lab Results on the UI, as well as Patients, Foods, and Dialysis Forms. However, I do not see any way for the clinician to see an interactive or general list of all the foods that are associated with a dialysis patient, and I believe this

information is critical as well. In general, we can add new entries on the UI to all tables other than that for the intersection table as I was unable to find anything for Patient_Foods on the UI itself. On the DML queries, I do see an instance where I can utilize FKs to associated added Lab Results with a unique patient or form, but I don't see that on 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).**

The INSERT does add the corresponding FK attributes, including at least one M:M relationship. The group has the appropriate INSERT statements that are features within the DML file and in the queries. I encountered INSERT for the Patient, Foods, Lab_Result, Patient_Foods and Dialysis_Forms entities of the database. They also do add up to the corresponding FK attributes as well. It looks great on the DML file, but I struggled to find this featured on the UI on the link that was provided for the front end.

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 is at least one DELETE, and the DELETE does remove things from a M:M relationship. In fact, there are several of these within DML file, and they are easily found. I can delete a patient, delete from foods – as well as form lab_results and dialysis_forms. There was another DELETE query written under "Other queries" where we are able to see how we would disassociate a person with a particular food. It cascades as expected when a patient or a certain food is deleted from their associated tables and there are delete icons features on the front-end UI.

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?**

After opening the queries, I do not see any queries for UPDATE. There is multiple SELECT queries for the page population and edit queries. I see that we can populate data with the SELECT statements, as well as getting a single patient, food, lab result, or dialysis – but unfortunately, I was unable to see at least one UPDATE for any one entity. I would not be able to tell if in the case of Foods, if amount, name, or calories could be updated for instance.

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.**

Upon reviewing the DDL file, I was unable to see a relationship that was NULLable, I could not find if there was at least one optional relationship. I believe that there are a few viable candidates where the relationship could be NULLable. One example of this would be to potentially have a relationship that is NULLable with Lab_Results or Patient_Foods entities. The justification for this is that the Lab Results can potentially be NULLable because the Patient could possible be continuing to receive lab tests but is considered no longer needing dialysis. In which case, we can provide a NULLable aspect to the primary key of dialysis_id for the Lab_Results entity.

The reason for Patients_Foods as being NULLable would be due to the patient not necessarily logging in food as actively on certain days where they may be distracted or away from any technology for instance. In this case, we can also have a NULLable component for the foreign key of Foods_food_id on the Patient_Foods entity, although it may be worth exploring having the same addition to the primary key of food_id on the Foods entity seeing that Patient_Foods is an intersection table.

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.

Good work overall! I enjoyed working with the UI and it was good to work with. It made sense and I was able to cross reference between the PDF document and DML/DDL files easily and seamlessly. I think an attribute that takes into consideration of the Dietary Preferences of the patient could be useful so that their selection of foods is better understood as well. There may also be room to feature an attribute of patients that are in critical need of kidney dialysis compared to others as I believe this could be helpful for the clinician when they are overwhelmed with patients at any given time.

Peer Reviewer #2: Clifford Bielinski

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 display them.

It seems almost every table is populated with a SELECT query and the results are displayed in the UI. However, I do not see a SELECT query or a place in the UI that displays the intersectional table of Patients_Foods. They do however have a SELECT query for finding specific foods a patient ate on a given day - although I don't really see where in the UI that information could be gleaned.

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

They have multiple SELECT queries that allow for search (by food_id, patient_id, etc) and a search bar at the top of the Food webpage that presumably will allow for dynamic display of content from the Foods table. I suspect they may need to modify the query so that it selects a single food by food.name rather than food_id since (presumably) the user will be searching by name in the search bar rather than by food_id.

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.

The UI has the ability to add new entries to each table except for the Patients_Foods table which (as mentioned above) I can't find on their web page. Each of these corresponds to proper INSERT queries in their DML file. I also can't find a place in the UI for them to associate new Lab Results with a specific patient or with a form of dialysis in their Add New Lab Result. Their INSERT queries however are properly formatted to include those attributes as FKs.

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).

They have the proper INSERT queries for Patients_Foods, which is their M:M relationship. Again, I just can't find that table in their UI but the syntax looks good in the DML file.

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 multiple delete queries and they seem to be formatted correctly. The DELETE syntax for the M:M relationship deletes by composite key. It appropriately cascades when either the food or patient is deleted from their respective entities. There are delete buttons on the UI for each table.

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?

I don’t see any UPDATE queries - they have SELECT queries for editing specific rows but the actual update queries seem to be missing for all the entities. The UI has EDIT buttons but I don’t see any forms to update existing table data (unless it just populates the data in the add forms below each table).

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.

From what I can tell I don’t see a NULLable relationship in their DDL file. Possibly Lab Results could have a nullable relationship with dialysis forms? For example maybe a patient is getting lab results but that patient is not yet on dialysis or no longer on dialysis post-transplant but are still getting labs. Then the dialysis_id in Lab_Results could possibly be made NULL in those instances.

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.

Overall it looks good! The UI is clean and easy to navigate. The attributes are well named and easy to understand.

Peer Reviewer #3: Anthony Beltran

- ***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 utilizes SELECT for every table in the schema.

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

Yes I saw a select to associate all the food given to the patient. I was looking for something simple(not simple to write the query) such as a query that wasn't a form that might select a patient with all the attributes. For example, a new doctor wants to see everything forms, diet, history on one particular patient.

- ***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 implements insert for every table in the schema. I noticed there were no table examples if a doctor wanted to see a list of all a certain patients foods.

I think this is wrong, but you get the idea?

select patients.name, patient.patient_id, foods.name

from Patients

join on Patients_foods where Patients_foods.patient_id = Patients.patient_id

join on Foods where Patient_foods.food_id = Foods.foods_id;

- ***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 inserts add the foreign keys.

- ***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 you had delete queries for every entity, including the M:M. There were no forms when I clicked the delete buttons. In your Create Tables, I didn't see any ON DELETE set to NULL for any of the tables. For example if a patient is deleted, the lab results are deleted of the patient. The results might be something you keep just in case you start tracking experimental drugs.

- ***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 is several UPDATES for multiple entities. There is no example forms for EDIT.

- ***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.**

You have several options to Null a relationship, but if you delete a person, it is not going to set the person's ID to null. I think you need the ON DELETE in your create tables.

- ***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.***

I think blood type is a good attribute for a patient. I also think it's important to rank the urgency for kidney donations. Some kidney patients are on the donor waiting list. Dr. House was one of my favorite characters when it was on T.V.

Feedback Received by Peer Reviewers- Summarized

Step I:

We had four peer reviewers for our project. The general consensus was that we did a great job showing the scope of the problem at large and the specific area/niche that our application/database would address. We were missing metrics on anticipated number of patients when rolling out the application. The feedback, from our reviewers,

also said we did a great job with presenting our entities within the overview. We also did well with conveying the different relationships between the attributes and constraints.

Our strongest part according to our reviewers was our attention to detail within our project outline when it comes to entity/attribute descriptions. We have clear 1:M and M:N relationships that are required for this project. We did not have an area for team assignments within the project. There were some minor details that should be adjusted to help maintain consistency between the entity relationship diagram and the written outline; which is that patients do have a M:N relationship with food but this relationship is implemented with food_id and patient_id as foreign keys inside Patients_Food. The Patients_Food table has attributes food_id and patient_id which should be prefixed as Foods_food_id and Patients_patient_id to match the convention used for other attributes.

Step II:

Our group spent a great deal of time on step II coming up with our own fixes and solutions prior to submitting the step II draft so we felt overall our draft was very strong to begin with. We had one peer reviewer and the feedback has been extremely positive. No concerns or suggestions were provided to fix at this time.

Step III:

We had three peer reviewers for part III. It was brought to our attention that the capitalization in DDL.sql file did not match exactly. It was suggested that we create a search tab to filter by food_id or food_name by two of the peer reviewers. Another concern that was brought up was that we needed the intersection table Patients_Food to be displayed on the UI. We were missing update queries in our DML.sql file. We also needed a nullable relationship. An idea to take into consideration is possibly adding dietary preferences of patients, blood types of patients, and urgency for kidney donations. We also received a suggestion that our UI associates new Lab Results with a specific patient or form of dialysis in their Add New Lab Result. Another suggestion was to include a SQL query that joins patients with foods. Finally, more feedback specified that we need dedicated forms for delete and edit in the UI for the Foods page.

Summary of Changes – Actions Based on Peer Review Feedback

Step I:

We decided to further expand on metrics for the anticipated number of patients we would target when rolling out the application. This is presented more in depth within our overview portion. We provided more references to support these anticipated numbers. We decided to fix the inconsistencies that were identified above with first fixing that patients do have a M:N relationship with food but this relationship is implemented with food_id and patient_id as foreign keys inside Patients_Food. The

Patients_Food table has attributes food_id and patient_id which we decided to prefix as Foods_food_id and Patients_patient_id to match the convention used for other attributes. In our project overview section, we also expanded on our long term goals for the project to capture feedback concerning prohibited foods for transplant patients.

We decided against actioning two pieces of feedback. First, we received feedback from several reviewers noting that we did not assign specific parts of the database implementation to team members. Although the question prompt addressed assigning specific sections to team members, this was not part of the rubric. We decided against adding this in order to encourage collaborative implementation of the entire database. Secondly, one suggestion was to remove the unit attributes from both the Foods and Lab_Results table, and to mandate a standard unit. We acknowledge that standard units will help for data analysis; however, we want to give our database flexibility to work with existing lab architecture, and so for the time being we want to include units as an attribute for phosphorus, potassium, and sodium. Additionally, we have the unit specified for each mineral as “NOT NULL”, which will help mitigate confusion about units used. We added in the table description recommended units.

Step II:

We only had one peer reviewer for step II. They provided very positive feedback. There were no suggestions of what to fix at this time. We spent a great deal of time working on step II and collaborating as a group. At this time, we see no other changes needed.

Step III:

We had a lot of feedback from our three peer reviewers for this part of the project. First, we rectified our capitalization in the DDL.SQL file to match the schema. We added a search tab in the UI to filter by food_id or food_name as this was suggested by two of our reviewers. Based on feedback, we added the Patients_Food intersection table into a separate page with basic sample data. We added update queries in the DDL.sql file. We also implemented two NULLable relationships between Lab_Results and Patients, and Lab_Results and Dialysis_Forms. Additionally, we corrected our foreign key constraints for our Patients_Foods intersection table to correctly delete entries when an associated patient or food was deleted.

We altered our UI for Lab Results to associate new Lab Results with a specific patient in their Add New Lab Result. We added some SQL queries using joins to display more data for each patient, ultimately hoping to create a ‘dash-board-view’ for a clinician on a specific patient. We created separate forms for delete and edit on the Foods page.

We were given three suggestions that ultimately we chose not to implement. First, we received the suggestion to add dietary preference as an attribute to patients. While we would like to consider this in the future, to keep our database simple, we are

currently focused on focusing on strictly pertinent nutrition information about end-stage renal disease patients. Secondly, we received a suggestion to make Patients_food NULLable with foods, as this could be used to show days when patients did not consume (or log) any foods. However, the project requirements state that we need to be able to delete items from a M:M relationship. Our only M:M relationship at this time is implemented between Foods and Patients via the Patients_Food intersection table. Instead, we added a query to our Data Manipulation Queries file to show all the foods that a patient consumed on a given day (resulting in an empty set if they logged nothing). Third, we received suggestions of adding blood type and urgency of kidney donations to our database. Again, while these are useful suggestions which we would like to implement in the future, we have chosen to focus on the implementation of the current database design for this stage of the project.

Summary of Changes – Upgrades to the Draft Version Step I

Our team collaboratively decided to implement some of the feedback provided above to this final step I version from our peer reviewers that were discussed within our Actions Based on Feedback section. Overall, we decided to act on the feedback that we felt was the most conducive/helpful to improve our draft except for the two areas noted in the previous section: team member assignments and the unit review suggestion. The reasons we did not include them can be reviewed within the previous section as well.

Changes we made on our own included we reviewed and fixed more grammatical errors/phrasing of paragraphs/outlines within our draft. We added more references to support how many potential patients this application/database would target. We reworded the first paragraph in the overview section. We found even more statistics to support our database/application.

Based on instructor feedback during office hours, we also eliminated a 1:M relationship between Dialysis_Forms and Patients, as this relationship was redundant with the 1:M relationship between Patients and Lab_Results,

We reviewed further our entities, attributes and constraints to make sure we were maintaining consistency. We fixed our intersection table based on information that we learned within this week's module. To ensure we maintained naming consistency, we changed the names of all attributes tracking units to their singular form. Also for naming consistency, we renamed our "Food" table to "Foods." We also double checked that all of our changes to the written outline matched the diagram. Additionally, we included each required section for this project and to make sure they were in the correct order. We also worked on normalizing our database to the Third normal form (3NF). We researched normalization and used normalizing principles to reduce the duplication of data, avoid data anomalies, and to ensure referential integrity which is found in a schema that follows the Third normal form such as our database because each of our tables uses a single primary key and each of our attributes uniquely depends on the

given row instance, our database meets the requirements for the 3NF. However, we learned we needed to address this further which is listed below.

Summary of Changes – Upgrades to the Step I Final Version/Step II Draft

First, we identified that we needed to include both an ER diagram and a schema to convey both detailed and generalized models for our database. We achieved this by re-creating our ER diagram with a simplified model, focusing only on relationships between entities, primary, and foreign keys. We then updated our schema based on normalization principles learned in the "Module 5: Normalization Steps" exploration.

We learned from this module that our database was not normalized as we thought from online research we conducted previously so we eliminated further redundant data such as units for each lab value and food nutrient units. We decided to utilize the Conventional Unit system for laboratory values which is adapting the standardized units used within the United States for these lab values: sodium mEq/L, potassium mEq/L, and phosphorus mg/dL.[1] Since they are based on the Conventional Unit system there is no need to explicitly label each of them in the database on the Lab_Results table. For food measurements we used the Metric system's standardized units, for example, potassium/phosphorus/sodium nutrient content would be in milligrams (mg), and calories would be in kcals within the Foods table. Food portions/amounts would be reported in grams on the Foods table.[2] We are using Customary units for weight and height in the Patients table which reports weight as pounds and height in inches.[3] We decided since we are using the typical standards that are reported within the dialysis community and in the United States that we decided to leave that information out of the database we designed.

Our feedback from our course grader was that we could potentially add a Doctors entity instead of the Dialysis_Forms entity table. We felt that the form of dialysis was important because of its impact on lab results based on whether the patient is on hemodialysis or peritoneal dialysis as there are different forms of dialysis to consider. The focus in this application is the dialysis patients so we kept the Dialysis_Forms entity but we added kidney_doctor information as an attribute within the entity table as we felt that this comment had value for the patients to help keep track of their kidney doctor. More details can be seen in the ER diagram, schema, and database outline sections.

---

1 *Conventional units – International Units*. GlobalRPH. (n.d.). Retrieved April 26, 2022, from https://globalrph.com/medical/conventional-units-international-units/

2 *Your guide to the New Food Label*. National Kidney Foundation. (2022, January 13). Retrieved April 26, 2022, from https://www.kidney.org/atoz/content/foodlabel

3 *What are customary units? - definition, facts and examples*. What are Customary Units? - Definition, Facts and Examples. (n.d.). Retrieved April 27, 2022, from https://www.splashlearn.com/math-vocabulary/measurements/customary-units

Summary of Changes – Upgrades to the Step II Final Version/Step III Draft:

We had no other feedback from our reviewers (peer and grader) about suggestions to fix for our step II draft. Our review that we received was extremely positive. We felt that we had a very strong part II so we left it the same.

Summary of Changes – Upgrades to the Step III Final Version:

We had quite a lot of feedback provided for our step III draft which we greatly appreciated. From our peer review feedback we fixed the following issues below:

We had a lot of feedback from our three peer reviewers for this part of the project. First, we rectified our capitalization in the DDL.SQL file to match the schema. We added a search tab in the UI to filter by food_id or food_name as this was suggested by two of our reviewers. Based on feedback, we added the Patients_Food intersection table into a separate page with basic sample data. We added update queries in the DDL.sql file. We also implemented two NULLable relationships between Lab_Results and Patients, and Lab_Results and Dialysis_Forms. Additionally, we corrected our foreign key constraints for our Patients_Food intersection table to correctly delete entries when an associated patient or food was deleted.

We altered our UI for Lab Results to associate new Lab Results with a specific patient in their Add New Lab Result. We added some SQL queries using joins to display more data for each patient, ultimately hoping to create a 'dash-board-view' for a clinician on a specific patient. We created separate forms for delete and edit on the Foods page.

We were given three suggestions that ultimately we chose not to implement. First, we received the suggestion to add dietary preference as an attribute to patients. While we would like to consider this in the future, to keep our database simple, we are currently focused on focusing on strictly pertinent nutrition information about end-stage renal disease patients. Secondly, we received a suggestion to make Patients_food NULLable with foods, as this could be used to show days when patients did not consume (or log) any foods. However, the project requirements state that we need to be able to delete items from a M:M relationship. Our only M:M relationship at this time is implemented between Foods and Patients via the Patients_Food intersection table. Instead, we added a query to our Data Manipulation Queries file to show all the foods that a patient consumed on a given day (resulting in an empty set if they logged nothing). Third, we received suggestions of adding blood type and urgency of kidney donations to our database. Again, while these are useful suggestions which we would like to implement in the future, we have chosen to focus on the implementation of the current database design for this stage of the project.

As a group we decided to remove the kidney_doctor attribute. It was suggested previously to include a doctor entity. We did not see this fit in our application and chose to keep the dialysis_forms. So we added kidney_doctor to the dialysis forms entity. However, upon reflection we realized that this attribute would not be relevant to the dialysis_forms entity. Additionally, normalizing our database with this attribute would require additional tables, resulting in additional complexity. Ultimately we decided to remove this attribute from our database design. We would consider adding a "primary care doctor" to the patients entity in the future though.

Project Outline and Database Outline:

Overview

Kidney disease impacts around 37 million people within the United States and is one of the leading causes of death. Health insurances pay an estimated $87.2 billion to treat chronic kidney disease and around $37.3 billion to treat those who have end-stage renal disease.[1] Around 81% of kidney disease patients own a cell phone that has the ability to connect to mobile applications.[2] More generations are tech savvy and becoming more reliant on telehealth/web development technologies. There are 786,000 patients with end-stage renal disease with appropriately 550,000 on dialysis within the United States.[3] We would anticipate that our application would start off with a goal of reaching 100,000 patients but a long-term goal of reaching the 81% of dialysis population with mobile phones that have application access.

The diet for individuals who have kidney disease is one of the most complicated diets to follow which include potentially restricting phosphorus, sodium, and potassium. There are currently more than 165,000 health-related and diet applications where only one-third of these actually focus on chronic diseases. Most of the current kidney-related applications information that is provided are not accurate or evidence-based. Many of them such as Kidney APPetite, Kidney Diet and Pocket Dietitian have been discontinued with newer operating systems.[4]

Our database driven website will provide a niche that is not currently being represented in today's market for dialysis. Especially bringing in renal dietitian skills paired with development; there is currently nothing that exists with this experience mixture but needs to in order to provide accurate and evidence-based information to those that struggle with their diet and require this information to be accessible to them. In this database, we will provide our Patient (on dialysis) with a tracking system for their Foods consumed which will allow the Patient to track specific nutritional content that is renal-focused to help them achieve their health goals/Lab Results that are within metric benchmarks. These Lab Results are influenced by what Foods the Patient decides to consume and impacted by the Dialysis Form they are on as well. This database will

provide the Patient with the resources ultimately to achieve better outcomes while tracking certain attributes that influence their abilities to achieve these goals.

Outside the scope of this project, we would look at including additional details, such as prohibited foods for transplant patients, or a limited record of what drugs patients may be taking. We also see this database as forming a structure that could be replicated to address other dietary-related health disorders.

Database Outline

Entities & Attributes:

Patients: Records the details of the patients. Focusing on patients with end-stage renal disease that are on dialysis.

- patient_id: INT (11), auto_increment, unique, not NULL, PK
- last_name: VARCHAR(128), not NULL
- first_name: INT(128), not NULL
- age: INT (11), not NULL
- gender: VARCHAR(20), default NULL
- height: INT(11), not NULL
- weight: INT(11), not NULL
- Using customary units for height and weight assumed to be reported as weight in pounds and height in inches.
- relationship: M:N relationship with Foods implemented with Patients_patient_id and Foods_food_id as a FK inside of Patients_Foods. Along with a timestamp of patient_food_time DATETIME().
- relationship: 1:M relationship with Lab_Results implemented with Patients_patient_id as a FK inside of Lab_Results.

Foods: records the foods the patients consume focusing on minerals/calories that need to be watched in the renal diet. These main minerals include phosphorus, potassium, and sodium. Along with focusing on calories consumed too. Recommended units for potassium, sodium, and phosphorus are mg, and the recommended unit for calories is kcals.

- food_id: INT (10), auto_increment, unique, not NULL, PK
- food_name: VARCHAR(128), not NULL
- phosphorus_content: INT(11), default NULL
- sodium_content: INT(11), default NULL
- calories: INT(11), default NULL
- potassium_content: INT(11), default NULL
- amount: INT(11), default NULL

- Using Metric system's standardized units: potassium/phosphorus/sodium nutrient content would be in mg, and calories would be in kcals. Food portions/amounts would be reported in grams.
- relationship: M:N relationship with Patients implemented with Foods_food_id and Patients_patient_id as a FK inside of Patients_Foods. Along with a timestamp of patient_food_time DATETIME().

Lab_Results: records the lab results of the kidney patients that are nutritionally relevant such as phosphorus, potassium and sodium levels. It is also important to monitor adequacy or also known as Kt/v which is a metric that shows how well the patient is dialyzing which is heavily influenced by the type of dialysis they are on since different forms of dialysis have different adequacy metrics to meet. Recommended units for phosphorus is mg/dL, and for sodium and potassium is mEq/L.

- lab_id: INT(11), auto_increment, unique, not NULL, PK
- phosphorus_lab: FLOAT
- potassium_lab: FLOAT, default NULL
- sodium_lab: INT(11), default NULL
- dialysis_adequacy_lab: FLOAT, default NULL
- lab_results_time: DATETIME(), default NULL
- Utilizes the Conventional Unit system for laboratory values which is adapting the standardized units used within the United States for these lab values: sodium mEq/L, potassium mEq/L, and phosphorus mg/dL.
- relationship: M:1 relationship with Patients implemented with Patients_patient_id as a FK inside of Lab_Results.
- relationship: M:1 relationship with Dialysis_Forms implemented with Dialysis_Forms_dialysis_id as a FK inside of Lab Results.

Dialysis_Forms: records the type of dialysis the kidney patient is on; common types of dialysis include in-center, home-hemo, and peritoneal dialysis. The form of dialysis can affect the desired lab results, and different forms of dialysis have different desired adequacy metrics.

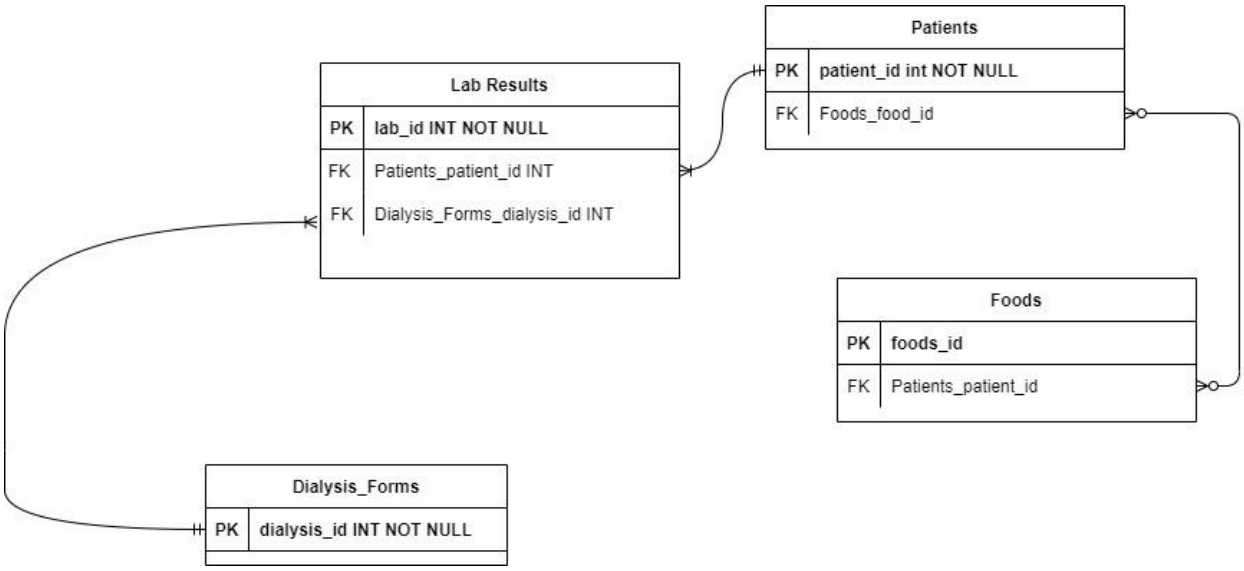
- dialysis_id: INT(10), auto_increment, unique, not NULL, PK
- name: VARCHAR(128), not NULL
- location_type: VARCHAR(128), not NULL
- adequacy_standard: FLOAT, not NULL
- relationship: 1:M relationship with Lab_Results implemented with Dialysis_Forms_dialysis_id as a FK inside of Lab_Results.

Intersection Table:

Patients_Food: Intersection table between Patients and Foods. This connects the patients to food that they have eaten, and adds a time characteristic for when the patient ate the food.

- food_patient_id: INT, PK
- Foods_food_id: INT (10), FK ON DELETE CASCADE
- Patients_patient_id: INT (11), FK ON DELETE CASCADE
- patient_food_time: DATETIME(), not NULL

Entity-Relationship Diagram:



---

[1] Centers for Disease Control and Prevention. (2022, February 28). *Chronic Kidney Disease Basics*. Centers for Disease Control and Prevention. Retrieved April 3, 2022, from https://www.cdc.gov/kidneydisease/basics.html#:~:text=15%25%20of%20US%20adults%20are,is%20about%2037%20million%20people.

[2] Singh, K. (2021, January 7). *Mobile Health in dialysis: The best engagement medium is the one that's with patients*. American Society of Nephrology. Retrieved April 14, 2022, from https://cjasn.asnjournals.org/content/16/1/12

[3] U.S. Department of Health and Human Services. (n.d.). *Kidney Disease Statistics for the United States*. National Institute of Diabetes and Digestive and Kidney Diseases. Retrieved April 14, 2022, from https://www.niddk.nih.gov/health-information/health-statistics/kidney-disease

[4] *Renal diet apps: Which one should I use?* Kidney Diet Tips. (2018, July 30). Retrieved April 3, 2022, from https://blogs.davita.com/kidney-diet-tips/renal-diet-apps-which-one-should-i-use/

Schema:

To normalize our schema we first created tables that we compiled that would fit our report structure in excel. We eliminated applicable redundant data to make it a first normal form (1NF). We then identified the primary keys within our tables and dependencies. Then we eliminated partial dependencies so that it fits the second normal form (2NF). We reassigned/omitted applicable dependent attributes. We then

ensured it was in the third normal form (3NF) by making sure we resolved all transitive dependencies. We followed the normalization steps that were in our module 5 exploration on normalization steps.[4]

Part of these normalization steps, we decided to utilize the Conventional Unit system for laboratory values which is adapting the standardized units used within the United States for these lab values: sodium mEq/L, potassium mEq/L, and phosphorus mg/dL.[5] Since they are based on the Conventional Unit system there is no need to explicitly label each of them in the database on the Lab_Results table. For food measurements we used the Metric system's standardized units: potassium/phosphorus/sodium nutrient content would be in mg, and calories would be in kcals within the Foods table. Food portions/amounts would be reported in grams on the Foods table.[6] We are using Customary units for weight and height in the Patients table which reports weight as pounds and height in inches.[7] We decided since we are using the typical standards that are reported within the dialysis community and in the United States that we decided to leave that information out of the database we designed.

By standardizing units with commonly used unit systems within the United States we were able to successfully decrease the amount of duplication that comes from our previous database and more so fitting the expected guidelines within a 3NF. It also eliminated the partial dependencies as well. See on next page the detailed schema model.

---

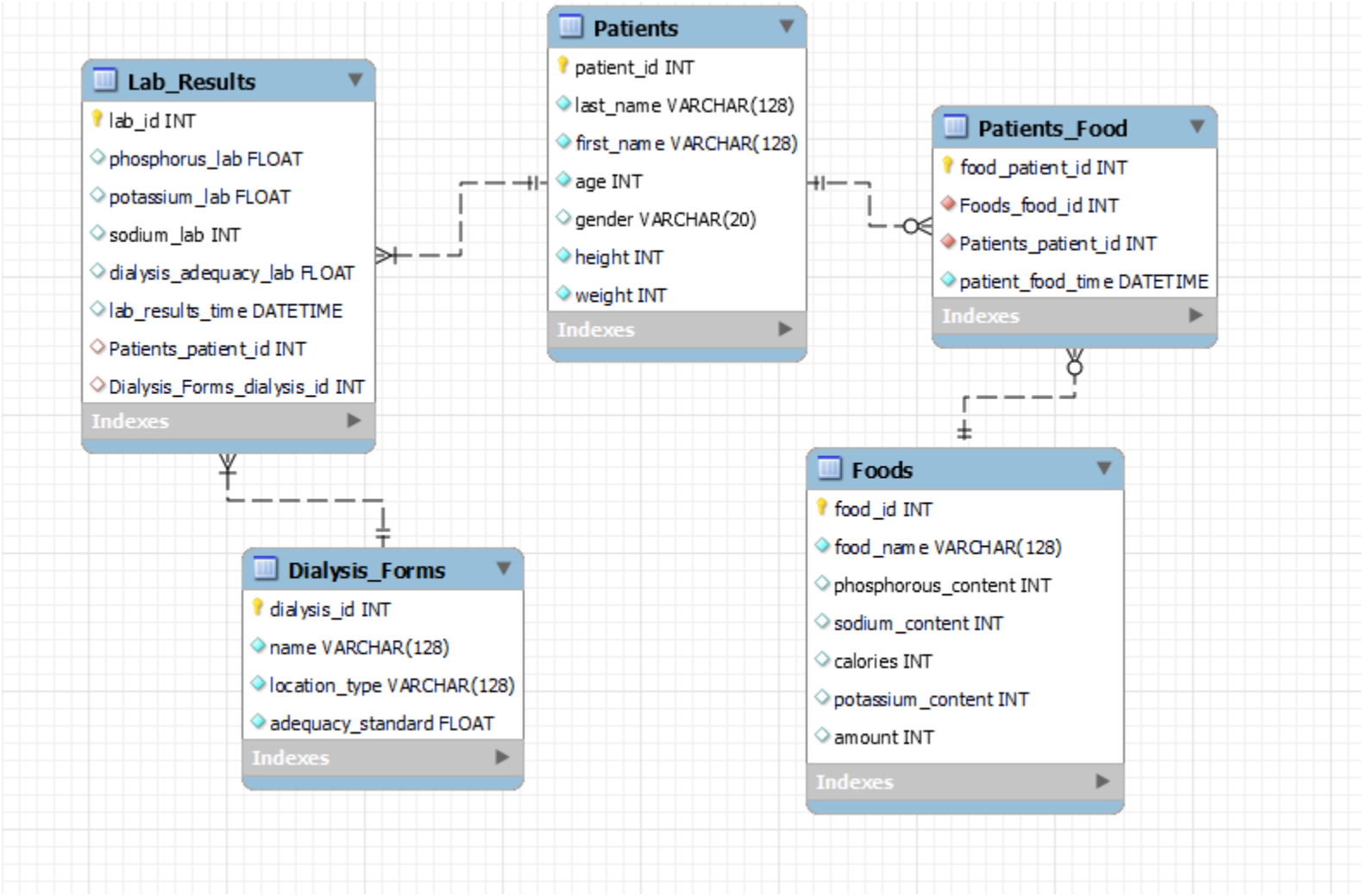
4 https://canvas.oregonstate.edu/courses/1870053/pages/exploration-normalization-steps?module_item_id=22036024

5 *Conventional units – International Units*. GlobalRPH. (n.d.). Retrieved April 26, 2022, from https://globalrph.com/medical/conventional-units-international-units/

6 *Your guide to the New Food Label*. National Kidney Foundation. (2022, January 13). Retrieved April 26, 2022, from https://www.kidney.org/atoz/content/foodlabel

7 *What are customary units? - definition, facts and examples*. What are Customary Units? - Definition, Facts and Examples. (n.d.). Retrieved April 27, 2022, from https://www.splashlearn.com/math-vocabulary/measurements/customary-units

Schema Model:



Example Data:

All sample data is based on fictional patients. Lab standards, dialysis forms, and food data are referenced from sources below. Refer as well to our SQL file containing our Data Definition Queries (DDL) for more information.

Patients Table:

		Patients				
patient_id	last_name	first_name	age	gender	height	weight
1	Smith	Arlene	55	F	64	145
2	Rogers	Christopher	63	M	72	180
3	Harrison	Kayla	68	F	65	125
4	Jackson	Henry	74	M	75	200
5	Wonders	Brenda	91	F	60	92

Patient information is fictional
patient_id = auto generating
Height in inches and weight in lbs considered Customary standardized units.

Lab_Results Table:[8]

	Lab_Results						
lab_id	phosphorus_lab	potassium_lab	sodium_lab	dialysis_adequacy_lab	lab_results_time	Patients_patient_id	Dialysis_Forms_dialysis_id
1	3.5	3.4	135	1.2	2022-05-07 23:22:05	3	1
2	5.5	3	142	1.7	2022-05-08 18:36:10	2	2
3	6.5	2.8	146	1.1	2022-05-01 20:20:06	4	1

8 *Understanding your lab work*. DaVita. (n.d.). Retrieved April 25, 2022, from https://www.davita.com/education/kidney-disease/symptoms/understanding-your-lab-work

4	10.5	6.6	144	0.6	2022-05-07 18:01:55	5	1
5	7.2	4.5	134	2.2	2022-05-11 10:19:25	1	2

lab_id, Patients_patient_id, Dialysis_Forms_dialysis_id = auto generating
Using Conventional standardized units for lab values: potassium and sodium measured in mEq/L and phosphorus in mg/dL. Adequacy is measured as a benchmark metric with hemodialysis achieving 1.2 and peritoneal dialysis is achieving 1.7.

Dialysis_Forms Table:[9] [10] [11]

		Dialysis_Forms		
dialysis_id	name	location_type	adequacy_standard	
1	hemodialysis FMC	incenter	1.2	
2	peritoneal Baxter	home	1.7	

Dialysis_id = auto generating
Adequacy is measured as a benchmark metric with hemodialysis patients achieving 1.2 and peritoneal dialysis 1.7.

---

9 *What is dialysis?* National Kidney Foundation. (2022, February 4). Retrieved April 25, 2022, from https://www.kidney.org/atoz/content/dialysisinfo

10 U.S. Department of Health and Human Services. (n.d.). *Peritoneal dialysis: Dose & Adequacy*. National Institute of Diabetes and Digestive and Kidney Diseases. Retrieved April 25, 2022, from https://www.niddk.nih.gov/health-information/professionals/clinical-tools-patient-management/kidney-disease/identify-manage-patients/manage-ckd/peritoneal-dialysis-dose-adequacy

11 U.S. Department of Health and Human Services. (n.d.). *Hemodialysis: Dose & adequacy*. National Institute of Diabetes and Digestive and Kidney Diseases. Retrieved April 25, 2022, from https://www.niddk.nih.gov/health-information/professionals/clinical-tools-patient-management/kidney-disease/identify-manage-patients/manage-ckd/hemodialysis-dose-adequacy#:~:text=The%20two%20methods%20generally%20used,blood%20flow%20through%20the%20dialyzer.

Foods_Table:[12]

		Foods				
food_id	name	amount	phosphorus_content	sodium_content	calories	potassium_content
1	Milk, whole	128	251	94.6	152	374
2	Beef, loin, top loin steak	284	585	128	423	801
3	Chicken, breast	174	419	81.8	275	597
4	Yogurt, Greek, nonfat	156	212	56.2	92	220
5	Kale	100	55	53	43	348

Food_id = auto generating
Units of measurement are based on the US metric system: phosphorus/sodium/potassium in milligrams(mg), calories as kcals, and amount/portion in grams.

Patient_Food Table:

		Patients_Food	
food_patient_id	Foods_food_id	Patients_patient_id	patient_food_time
1	5	1	2022-05-10 15:40:11
2	4	2	2022-05-20 18:32:04
3	3	3	2022-05-15 12:08:12
4	2	4	2022-05-11 15:07:55
5	1	5	2022-05-16 10:22:28

12 *Fooddata Central Search Results*. FoodData Central. (n.d.). Retrieved April 25, 2022, from https://fdc.nal.usda.gov/fdc-app.html#/

Foods_food_id and Patients_patient_id = FKs referencing Foods and Patients tables.