CS 340 Group 35: Pho Dac Biet
Project Step 5 Draft

Team Members: Dan Truong, Charles Nguyen
Project Title: Pharmaceuticals Database
Project URL: http://flip2.engr.oregonstate.edu:17350/index

Overview

"No Refills" Pharmacy is a single pharmacy that dispenses over 500 C2 (Schedule II controlled substance) prescriptions (which are not refillable) from a collection of over 250 controlled substances on a day-to-day basis. To meet this demand, the pharmacy orders C2 controlled substances from 5-10 different distributors. Dispensing C2 prescriptions is solely completed by pharmacists, and "No Refills" utilizes about 15 dedicated pharmacists for this purpose.

In order to comply with DEA (Drug Enforcement Administration) regulations, a perpetual inventory must be kept and maintained for all controlled substances entering and exiting the pharmacy. A database driven website would allow the pharmacy to effectively keep track of its inventory (*Drugs*), dispensing records (*Prescriptions*) by its employees (*Pharmacists*), and ordering records (*Invoices*) from its suppliers (*Distributors*).

Database Outline

Pharmacists: This records the details of the Pharmacists who are dispensing Prescriptions.

- rph_id: INT, auto-increment, not NULL, PK
- rph_first_name: VARCHAR(45), not NULL
- rph_last_name: VARCHAR(45), not NULL
- ytd_rxs_processed: INT, not NULL, default 0
- 1:M optional relationship between Pharmacists and Prescriptions. rph_id PK implemented as rph_dispensing FK in Prescriptions.

Team member: Dan

Prescriptions: This records the details of the Prescriptions of Drugs dispensed by Pharmacists.

- rx_number: INT, auto-increment, not NULL, PK
- rph_id: INT, FK
- rx_date: DATETIME, not NULL
- drug_ndc: BIGINT, not NULL, FK
- units_dispensed: INT, not NULL
- rx_price: DECIMAL(8,2), not NULL
- M:1 relationship between Prescriptions and Pharmacists. rph_id FK sourced from rph_id PK in Pharmacists.
- M:1 relationship between Prescriptions and Drugs. drug_ndc FK sourced from drug_ndc PK in Drugs.

Team member: Dan

Drugs: This records the details of Drugs, from Invoices, dispensed as Prescriptions.

- drug_ndc: BIGINT, unique, not NULL, PK
- drug_name: VARCHAR(45), not NULL
- strength_in_mg: INT, not NULL
- form: VARCHAR(25), not NULL
- is_generic: TINYINT(1), not NULL, default(1) is generic
- manufacturer: VARCHAR(25), not NULL
- units_in_stock: INT, not NULL, default 0
- 1:M optional relationship between Drugs and Prescriptions. drug_ndc PK implemented as drug_ndc FK in Prescriptions.
- M:M relationship between Drugs and Invoices. drug_ndc PK implemented as drug_ndc FK in Drugs_Has_Invoices.

Team member: Dan

Invoices: This records the details of Invoices representing the ordering of Drugs from Distributors.

- po_number: INT, auto-increment, unique, not NULL, PK
- distributor_number: INT, FK
- order_date: DATETIME, not NULL
- order_cost: DECIMAL(8,2), not NULL
- M:M relationship between Invoices and Drugs. po_number PK implemented as po_number FK in junction Drugs_Has_Invoices.
- M:1 relationship between Invoices and Distributors. distributor_number FK sourced from distributor_number PK in Distributors.

Team member: Charles

Drugs_Has_Invoices: This intersection table records the details of Drugs ordered as Invoices.

- drug_ndc: BIGINT, not NULL, FK
- po_number: INT, not NULL, FK
- units_ordered: INT, not NULL
- drug_subtotal: DECIMAL (8,2), not NULL
- M:1 relationship between Drugs_Has_Invoices and Drugs. drug_ndc FK sourced from drug_ndc PK in Drugs.
- M:1 relationship between Drugs_Has_Invoices and Invoices. po_number FK sourced from po_number PK in Invoices.

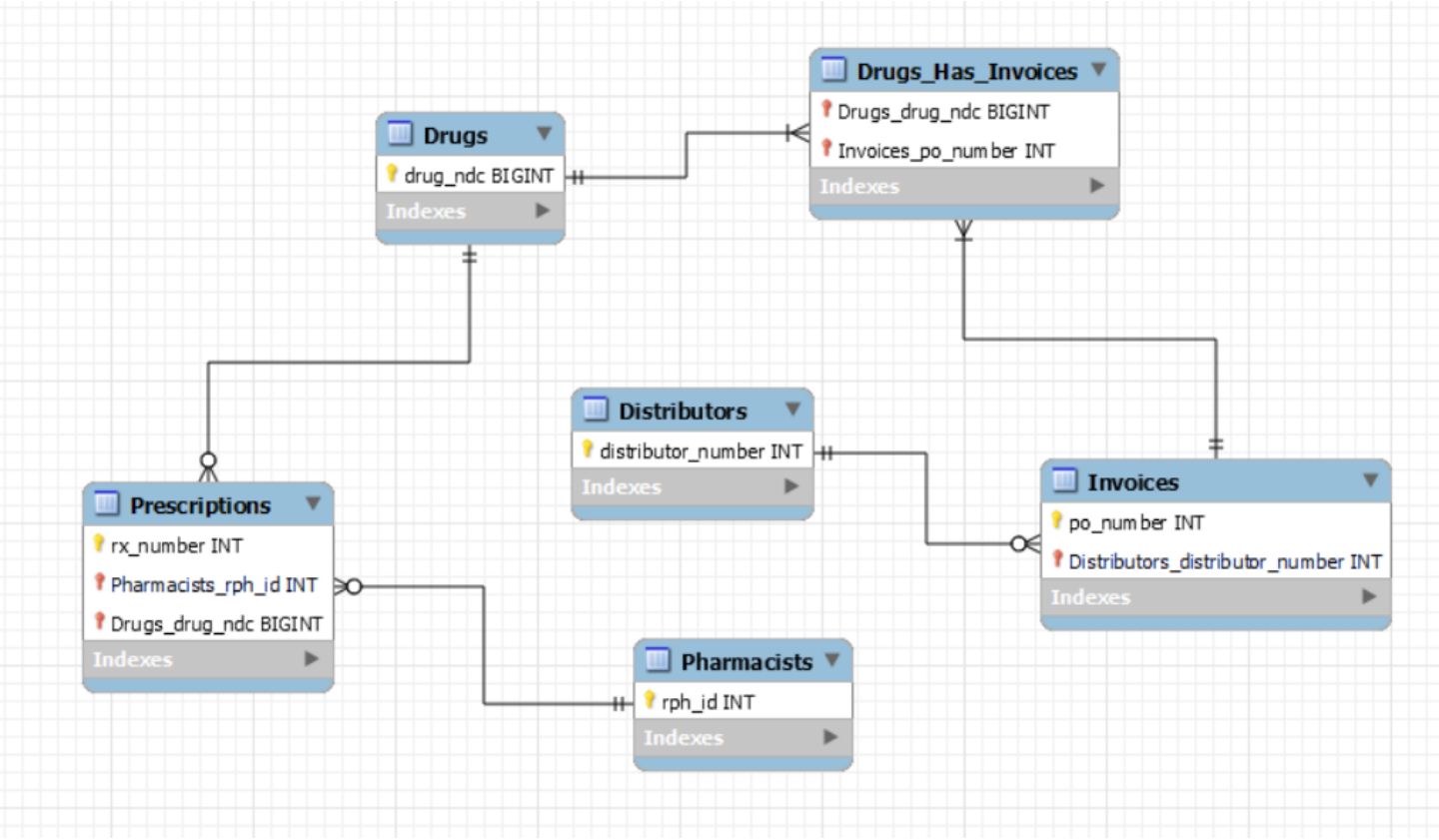
Team member: Charles

Distributors: This records the details of Distributors through which orders are placed as Invoices.

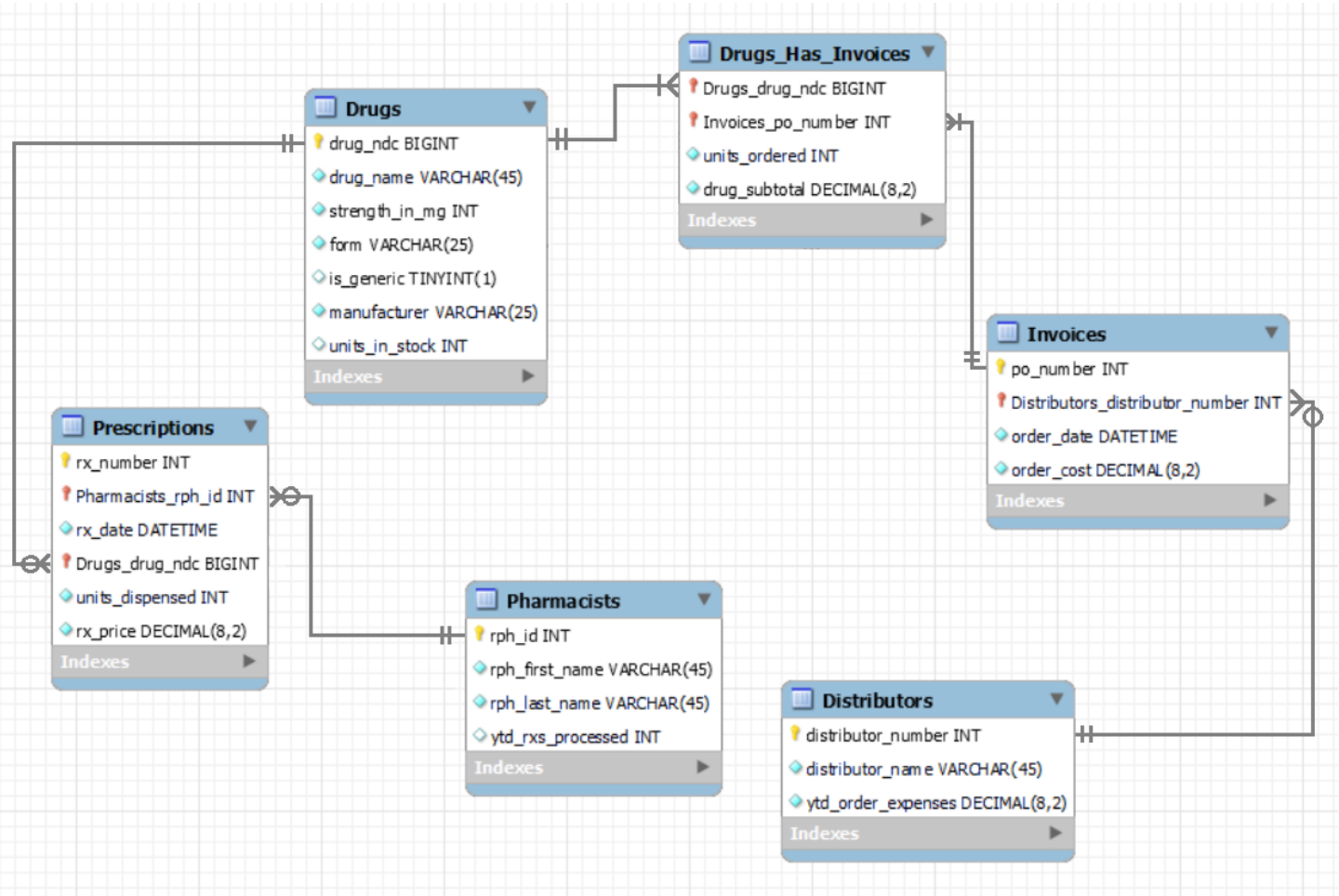
- distributor_number: INT, auto-increment, unique, not NULL, PK
- distributor_name: VARCHAR(45), not NULL
- ytd_order_expenses: DECIMAL(8,2), not NULL
- 1:M optional relationship between Distributors and Invoices. distributor_number PK implemented as distributor_number FK in Invoices.

Team member: Charles

ER Diagram



Schema



Sample Data Tables

Pharmacists

rph_id	rph_first_name	rph_last_name	ytd_rxs
1	Mark	Chee	2
2	Sora	Kim	0
3	Brian	Lu	1
4	Long	Phan	1
5	Yueshi	Lin	1

Prescriptions

rx_number	Pharmacists.rph_id	rx_date	Drugs.drug_ndc	units_dispensed	rx_price
1	1	4/1/2022	54092038701	30	$10.00
2	1	4/2/2022	00406889201	60	$5.47
3	4	4/3/2022	59417010410	30	$30.00
4	5	4/3/2022	16714082201	28	$4.25
5	3	4/3/2022	00228306001	14	$2.37

Drugs

drug_ndc	drug_name	strength_in_mg	form	manufacturer	is_generic	units_in_stock
00228306001	Amphetamine Salts ER	20	capsules	Actavis	1	986
00406889201	Amphetamine Salts	10	tablets	Mallingkrodt	1	940
16714082201	Methylphenidate	10	tablets	Northstar	1	972
54092038701	Adderall XR	20	capsules	Shire	0	970
59417010410	Vyvanse	40	capsules	Shire	0	970

Drugs_Has_Invoices

Drugs.drug_ndc	Invoices.po_number	units_ordered	drug_subtotal
00228306001	2	1000	$185.45
00406889201	2	1000	$421.32
16714082201	2	1000	$319.76
54092038701	1	1000	$314.15
59417010410	3	333	$196.59
59417010410	4	667	$393.20

Invoices

po_number	Distributors.distributor_number	order_date	order_cost
1	3	3/1/2022	$314.15
2	2	3/11/2022	$926.53
3	1	3/21/2022	$196.59
4	1	3/31/2022	$393.20

Distributors

distributor_number	distributor_name	ytd_order_expenses
1	McKesson	$589.79
2	AmerisourceBergen	$926.53
3	Cardinal	$314.15

Updates in Step 5:

There was a bit of feedback regarding CSS related topics or minor tweaks to make the pages look nicer. We decided we would not look into these for Step 5, as we are focused on implementation for the time being. However, they are things we will look towards before the final submission as time permits. There was also feedback in regards to SQL running very long in the Python file. I am currently using word wrap, which prevents the issue for me personally, and it has been a struggle to get it to work properly while manually wrapping the SQL. It will be something I look towards fixing before the final submission. Outside of feedback, significant changes were made to DML.sql as a result of implementation of further operations. Many of these were associated with updating values in other tables based upon updates in current ones being worked with. CRUD operations should be functional for all tables, at least as far as implementations were intended.

Feedback from Zachary Smith for Step 4:

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

The CRUD functions implemented for Pharmacists and Distributors work correctly.

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

I really like the overall layout and design of the project. On the inventory home page, y'all could space out the date, RPH, Quantity, and Price so that they line up. That's a really minor detail but something I noticed.

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

As others have mentioned some of the links are un-navigatable when clicked on. I looked at your app.js file and it just seems like you guys don't have the routes set up for them yet.

This is off topic but I thought it was kind of funny the only drugs loaded into the inventory are stimulants, being that we are in college in everything. As someone with ADHD I instantly recognized all of them and had a little laugh. Also thought the generic checkbox was a good addition, as it's a minor detail in pharmaceuticals but makes a big difference in like pricing and stuff.

Feedback from Edgar Palaquibay for Step 4:

Great job! After implementing a few CRUD operations it gets easier for future entities. Looking forward to seeing the final product.

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

The stated entities, Pharmacists and Distributors work for all CRUD operations.

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

The UI is easy to navigate and use, I would suggest to move everything after the RPH dropdown in C2 Inventory to the next line. The other entities look great and have no other UI suggestions. I am wondering though what would happen if I entered a new drug with the generic box checked.

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

Only blocked due to timing, so no worries there.

Feedback from Jacob Hathaway for Step 4:

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

Crud steps worked great in all cases. No errors during inputs. Some minor improvements suggested below.

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

One minor nitpick is that on the edit screen each value has to be entered again. I like the idea of placeholder text, but sometimes I just may want to edit only one field without retyping everything.

Recommend including a deletion confirmation alert of some kind, just in case the user was spilling coffee and grabbed the mouse and accidentally hit delete.

Title sizing is a little off. Main title is very small compared to the table title. Recommend looking at that.

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

It looks like your j2 templates need a 'base' to extend from. For example, I recommend you create a base.html template that has your navigation, a block title macro and maybe a block content macro. Then you can "extend" your template into your other pages to replace the excess HTML in each of your j2 templates.

Recommend taking your SQL queries and placing them into an external python file and importing them into your main app.py file. This will help clean up your routes. Looks like the SQL queries are running really long to the right in the file which makes them difficult to read/maintain. To break them up you could do something like below. Then you could just call your function to get the resulting string.

```
def mySpecialQuery():
   myQuery = "SELECT * FROM myTable WHERE "
   myQuery += "myThing = 1;"
   return myQuery
```

Feedback from Annie Wan for Step 4:

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

Pharmacists and Distributers have full CRUD implementations, and they work fabulously and refresh immediately. Nice work!

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

The UI is straight-forward and specifies the input domains for the different values (e.g. for adding a Strength value on the Inventory page, it is shown that the value can only be a number).

I myself know nothing about pharmacies and drugs, and so I was not sure what specific values I could make up to add into the Inventory page, but that's not relevant. The database admin would be able to enter/edit/delete data appropriately without a problem.

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

As Dan and Charles mentioned, the Drug Details, Drug Search, Invoices, Invoice Details, and Prescriptions links don't navigate anywhere at the moment. I'm able to view the Inventory page but I'm unable to add a drug. I'm sure Dan and Charles will implement these fully when they get enough time.

Updates in Step 4:

We changed the distributor_number FK attribute of the Invoices entity to no longer be NOT NULL. This allows us to set the FK to ON DELETE SET NULL, setting the attribute to NULL in the case that a distributor_number PK is deleted. The DDL.sql was updated accordingly.

DML.sql was adjusted in a few places to accommodate to how the web application ended up being built.

The following query was added to the index page in order to facilitate a dropdown menu to select an RPH:
"SELECT rph_id, CONCAT (rph_first_name, ' ', rph_last_name) AS 'RPH' FROM Pharmacists Order BY rph_first_name;"

The following query to display all Distributors for the distributors page was amended to facilitate displaying distributors with NULL SUM(Invoices.order_cost), with these values being set to 0:
"SELECT Distributors.distributor_number AS 'Number', Distributors.distributor_name AS 'Distributor Name', IFNULL(SUM(Invoices.order_cost), 0) AS 'Total Expenses' FROM Distributors LEFT JOIN Invoices ON Distributors.distributor_number = Invoices.distributor_number GROUP BY Distributors.distributor_number;"

The following query to display all Pharmacists for the pharmacists page was amended to include the rph_id primary key:
"SELECT Pharmacists.rph_id, Pharmacists.rph_first_name AS 'First Name', Pharmacists.rph_last_name AS 'Last Name', COUNT(Prescriptions.rph_id) AS '# of C2s Processed' FROM Pharmacists LEFT JOIN Prescriptions ON Pharmacists.rph_id = Prescriptions.rph_id GROUP BY Pharmacists.rph_id;"

The following query was added to Pharmacists to select a specific Pharmacist for updating:
"SELECT * FROM Pharmacists WHERE rph_id = %s;"

Fixes based on feedback from peers for Step 3:

"Yes, all tables have a select. Great work on this by the way. Everything leads well i nto each other. Invoices leads to Transactions from the table. I would say maybe make Transactions available on the navigation bar as well."
Added the Transactions to navigation bar as "Invoice Details". The Invoices html page now the Invoices and Drugs_Has_Invoices tables. The new transactions.html page will display the specific transactions for the selected invoice and allow user to make changes to the Drugs_Has_Invoices table.

"There is a UI element for a search function, but I cannot find anything in the DML.sql file that could suggest a search to populate a table. I figure these would usually be denoted by a Join, and there are Join statements that exist within the DML, but none that are commented to be denoted to serve the search function."
Currently, because the current html pages have limited functionality and only serve as an example, the drug search UI is supposed to lead into the drug audit webpage that displays all prescriptions and invoice records for that specific drug. We do have the data manipulation queries for this.

"Is at least one relationship NULLable? No, looking over their schema and the options for deletion in the UI, I believe every relationship is required among every table. They have stated that the relationship between Invoices and Distributors is optional, and yes, Distributors can exist without Invoices, but if the relationship is already established, deleting distributors from Invoices would not work since it is a Foreign Key."
In response to this, we have changed our rph_id foreign key in our Prescriptions entity to not be required and used the ON DELETE SET NULL modifier.

"The bottom of some letters in your NavBar seem to be clipping (characters like y and g are sliced making y look like v and g look like a). After inspecting your elements, the most likely culprit I could think of is that the tag's padding is too low. If that does not work, I suggest fiddling w ith the CSS of the tag's margins and height until the clipping goes away."
We fixed the tag in the style.css to allow for extra space and our navigation bar no longer clips.

Feedback from Russ McCuen for Step 3:

Does the UI utilize a SELECT for every table in the schema?
Yes, the UI utilizes a SELECT for every table in the schema.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?
Yes, on the drugsearch.html page there is the ability to search by NDC.

Does the UI implement an INSERT for every table in the schema?
Yes, the UI implements an INSERT for every table in the schema. I have a suggestion on this section below in the final question prompt.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?
Yes, each INSERT also adds the corresponding FK attribute. The M:M relationship is between *drugs* and *invoices.*

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?
Yes, there is at least one DELETE. The UI has options to delete from *Prescriptions, Pharmacists, Invoices,* and *Distributors*. The M:M requirement is fulfilled by the DELETE option in *drugs_has_invoices*.

Is there at least one UPDATE for any one entity?
Yes, there is at least one UPDATE for any one entity. All of the pages that have a DELETE option also have an UPDATE option (listed as **edit**) on their respective page. These pages are *Prescriptions, Pharmacists, Invoices,* and *Distributors*.

Is at least one relationship NULLable?
Opinion seems split on this one based on reviews but I'm going to say that, yes, at least one relationship is NULLable. If you look at the draft, the table for **Pharmacists** has one pharmacist, Sora Kim, who has filled 0 ytd_rxs. The prompt for this question states, "there should be at least one optional relationship, e.g. having an Employee might be optional for any Order." In this table you can see that most pharmacists have a value > 0 is ytd_rxs, but not all do. Because of this, I think the NULLable relationship requirement is fulfilled.

Do you have any other suggestions for the team to help with their HTML UI?
This is purely personal preference: I like that you have a current page display on the nav bar, but it makes the bar move in between pages. Instead of this, maybe change the font color of the current page so the nav bar itself remains static.

I agree with a previous review that there should be the ability to add a prescription on the *prescriptions* page.

Also, **No Refills Pharmacy** is clickable on every page, but it is linked to the current page. I do not know if this was meant to be a static link to the homepage, but right now the ability to click on it doesn't seem to provide any practical function.

Feedback from Ian Craig McMillan for Step 3:

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.**
Mostly, but I didn't see the intersection table Drugs_Has_invoices on the UI.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?
Yes, the team went above and beyond and implemented an actual search function that happens to satisfy the requirements by the nature of its SQL properties. Referencing discussion post #609 all that was meant by this requirement is to use joins to display relevant data in a table. For example when displaying the pharmacists they used a join to also display the prescription count. This would also satisfy the requirment.

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.**
I think so, but without testing it I can't quite see how I would do an insert into the intersection table.

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).**
I don't think the explanation of this question is sufficient to cover the differences between everyones projects and does not apply in all cases. In this project, they have included the FK in all INSERTs, and have included a method to insert into the M:M relationship. So I believe the requirement is satisfied.

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.**
Many DELETES are implemented but I believe that to satisfy this requirement ON DELETE CASCADE should be used. but without testing it I am not 100% sure.

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

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

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.
Not really, the only thing I would say is give a double check to if you need to implement some ON DELETE conditions or not.

Feedback from Katie Schaumleffle for Step 3:

Does the UI utilize a SELECT for every table in the schema?
Yes, it has an easy to use interface. The navbar at the top makes it easy to go between the different tables. I might suggest making the current tab bold instead of writing “current”, but that is minor and more of an aesthetic suggestion than anything for this assignment.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?
Yes, under the “Drug Search” tab, there is a way to search a list of drugs by NDC. I do see the use of SELECT in your DML.sql file as well which correlates to this search.

Does the UI implement an INSERT for every table in the schema?
Yes, there are elements in the UI to add new data for each table in the schema, however it is a little confusing to have to leave the prescription page and go to inventory to add a new prescription. If a doctor would like to add a new prescription for a patient, I would suggest having that under the prescription page.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?
Yes, it appears that all of the INSERT statements that have a FK make the correct references when needed. The M:N relationship between drugs and invoices also make use of the INSERT statements as well.

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?
Yes, there are several delete buttons throughout this project. The Drugs_has_invoices is the intersection table for the M:N relationship, which also has a DELETE statement.

Is there at least one UPDATE for any one entity?
Yes, each table has an element allowing the user to edit their data. The DML.sql file matches this and also includes the UPDATE queries.

Is at least one relationship NULLable?
The outline states there is an optional relationship between Pharmacists and Prescriptions, however this doesn't seem like it'd work since rph_dispensing is the FK from Prescriptions being used here. Based on the outline and the UI, it does not appear that there are any NULLable relationships.

Do you have any other suggestions for the team to help with their HTML UI?
Overall, your UI is looking good! As I suggested earlier, I would suggest making the current tab in bold, rather than writing "current tab" next to it. Also, the "No Refills Pharmacy" button at the top of each page doesn't take you anywhere. I would assume it would take me back to the home page, but it just reloads the current page. I would suggest either not making this a button, and just make it a heading at the top of the page, or have it direct the user home.

Feedback from Danny Chung for Step 3:

Does the UI utilize a SELECT for every table in the schema?
Yes, all tables have a select. Great work on this by the way. Everything leads well into each other. Invoices leads to Transactions from the table. I would say maybe make Transactions available on the navigation bar as well.

Does at least one SELECT utilize a search/filter with a dynamically populated list of properties?
There is a UI element for a search function, but I cannot find anything in the DML.sql file that could suggest a search to populate a table. I figure these would usually be denoted by a Join, and there are Join statements that exist within the DML, but none that are commented to be denoted to serve the search function.

Does the UI implement an INSERT for every table in the schema?
Yes, the UI has an element to add new values to a database for every table in the schema.

Does each INSERT also add the corresponding FK attributes, including at least one M:M relationship?
Yes, each INSERT statement that has a foreign key makes the correct select statements to make a reference whenever a foreign key is expected.

Is there at least one DELETE and does at least one DELETE remove things from a M:M relationship?
Yes, most tables offer a delete button, and there is a M:M relationship (Drugs_has_invoices) in the schema, Transactions.html in the website. It allows for deletion. Where deleting this would not delete any Distributors or Drugs.

Is there at least one UPDATE for any one entity?
Yes, as a matter of fact, all tables have a UI element that offers an edit button, but there does not appear to be a form of any sort to fill forms pertaining to the update.

Is at least one relationship NULLable?
No, looking over their schema and the options for deletion in the UI, I believe every relationship is required among every table. They have stated that the relationship between Invoices and Distributors is optional, and yes, Distributors can exist without Invoices, but if the relationship is already established, deleting distributors from Invoices would not work since it is a Foreign Key.

Do you have any other suggestions for the team to help with their HTML UI?
The bottom of some letters in your NavBar seem to be clipping (characters like y and g are sliced making y look like v and g look like a). After inspecting your elements, the most likely culprit I could think of is that the <ul> tag's padding is too low. If that does not work, I suggest fiddling with the CSS of the <ul> tag's margins and height until the clipping goes away.

Fixes based on feedback from peers for Step 2:

"Attributes are mostly in the singular. In the Drugs entity, "units in stock" somewhat implies plural. Suggestion: This could be changed to quantity_of_stock or just quantity. In Pharmacists entity, ytd_rxs_processed also somewhat implies plural. This could be changed to total_rxs_processed_ytd. Likewise the in Distributors entity, ytd_order_expenses could be changed to total_order_ytd." - John Wong

While both reviewers mentioned this as an issue, we decided to not make changes to our database for the time being because we feel that an amount signifies a singular attribute. However, we will take this feedback into consideration going forward depending upon how our knowledge of appropriate syntax changes.

"The schema is not that easy to read because although the FK are present, they don't share a consistent naming convention with the primary key and it takes a few looks to see which correspond to which. You can show the relationship of the FKs to PKs in the schema easier by generating it in the designer tab in phpMyAdmin. Something like this." - John Wong

Both reviewers made a similar comment regarding this issue as well. We changed our FK to mirror their respective PK for the time being. These include Pharmacists entity FK attributes rph_id and drug_ndc, Drugs_Has_Invoices intersection table FK attributes drug_ndc and po_number, and Invoices entity FK distributor_number. ERD, Schema, example tables, and SQL were updated accordingly.

"The primary and foreign keys are correctly defined. Some consistency in how foregin keys are named would be appreciated. The ON DELETE CASCADE is implemented on the Drugs_Has_Invocies and is declared correctly." - John Wong

FK were adjusted to match PK as per their previous advice.

"The schema is easy to read, and there are no crossed lines. However, I may be mistaken, but I believe the course staff may consider the Schema as presented to be simply a well fleshed out ERD. It is my understanding that the ERD should focus on the relationships (1:M, M:N, etc) while the Schema should focus on how the information is connected. That information is technically present if you can follow along with the PK-FK relationships, but since you renamed them all when inserting them as FKs that information is lacking. I think the staff may be looking for lines actually connecting the PK to the FK as shown in the example schema we downloaded in Advanced SQL assignment." - Ian Craig McMillan

We adjusted the Schema to have relationship lines which point directly from PK to FK. We also ensured that PK and FK were given consistent naming, as per previous advice.

"There are no repeating rows, but each value is not atomic therefore 1NF is NOT satisfied. If rph_name is broken down to first and last name, 1NF will be satisfied. For 2NF I think you might have a partial dependancy in the Prescriptions table. If I understand correctly, rx_price is dependent on drug_dispensed and unit dispensed. Therefore it might make more sense to just have price per unit of each drug in the Drugs table. I am not 100% sure on this one, but it is worth investigating." - Ian Craig McMillan

The two reviewers were in disagreement over whether 1NF would failed to be satisfied by rph_name not being broken down into first name and last name. We decided to separate them into rph_first_name and rph_last_name just to be sure. Schema, tables, and SQL were updated accordingly.

We did not adjust rx_price to price_per_unit as suggested, as rx_price would not necessarily be a function of drug_ndc and units_dispensed, as price per unit quantity could change depending on the quantity dispensed.

"The keys are all correctly defined as per the schema. However, I really don't see the point in renaming the keys from their PK to FK roll. for example, in the Invoices table you have put: "FOREIGN KEY (order_distributor) REFERENCES Distributors(distributor_number)" Why not just use: "FOREIGN KEY (distributor_number) REFERENCES Distributors(distributor_number)" - Ian Craig McMillan

We adjusted these accordingly as per previous feedback.

“Since a Prescription can't exist without a Pharmacist and a Drug, I think you need to add ON DELETE CASCADE to the drugs and pharmacist FKs in Prescriptions. you have one ON DELETE CASCADE in your intersection table Drugs_Has_Invoices, but I think you should have a ON DELETE CASCADE for both FKs.” - Ian Craig McMillan
We adjusted the SQL to include ON DELETE CASCADE for drug_ndc FK of the Drugs_Has_Invoices intersection table, as well as rph_id FK and drug_ndc FK of the Prescriptions entity.

Fixes based on feedback from TA for Step 1:

“Drugs_has_Invoices does not have relationships listed. It should have a 2 Many to One to Invoices and Drugs. It is on Prescription but missing on Drugs_has_Invoices. For Drugs_has_Invoices, please make capitalization more consistent for next time! (ie. Drugs_Has_Invoices or Drugs_Invoices).”
We updated the database outline to include relationship information for Drugs_Has_Invoices, indicating that it has two M:1 relationships with the Invoices and Drugs entities. We also updated the name of the entity Drugs_has_Invoices to Drugs_Has_Invoices for more consistent capitalization.

We also chose to make changes to the Drug entity. We added attributes for form (capsule/tablet) and manufacturer, and we changed the strength attribute name to strength_in_mg for clarity as far as units of measurement. We adjusted the Drugs entity PK drug_ndc from INT to BIGINT to accommodate for the length of NDCs and changed the corresponding FKs drug_dispensed in the Prescriptions entity and ordered_drug_ndc in Drugs_Has_Invoices from INT to BIGINT accordingly.

Fixes based on feedback from peers for Step 1:

"I would like to see how many distributors there are and roughly how many employees there are in each location."
We adjusted the overview section to clarify that this is a single pharmacy, including the number of pharmacist employees assigned to dispensing C2 drugs as well as the number of distributors.

“I don't think you should have the RX number as the PK for prescriptions, since you keep the same RX every time you refill. So that wouldn't make the PK unique. You could probably combine the RX with the date/time to get a unique PK.”
We did not act upon this feedback, as C2 prescriptions are not refillable by law, thus requiring a unique prescription each time they are filled. We adjusted the overview accordingly to briefly explain Schedule II (C2) drugs.

In response to whether each team member is assigned to associated pages: **“It doesn't indicate which team member...”**
We assigned each entity to a specific team member for primary implementation accordingly.

“In drugs, I don't think you need to keep track of cost per unit. Because this can be calculated quickly on summary page. ”
We added a drug_subtotal attribute under the Drugs_has_Invoices junction/entity and removed the price_per_unit attribute from the Drugs entity. The ERD was fixed accordingly.

“Same concept under prescriptions with profit. That can be quickly calculated, but I would rather see the price paid for the prescription.”
We changed the profit attribute of the Prescription entity to rx_price, denoting the price paid for the prescription. The ERD was fixed accordingly.