FtDbT Project

This is the presentation for Sean Fletcher, Aimee Haas and Nick Largey’s Part 2 of COS457’s database project entitled “Farm to DataBase Table”. Our goal is to create a database system that simplifies the process of getting locally grown, healthy produce and meats into public schools. We are accomplishing this by creating a database system that provides a simple and direct user interface, handling a majority of traditional roadblocks such as locating local producers, temporary storage to ease logistic and delivery scheduling complications, automatic purchase order creation for accounting and data analytics and Volunteer staffing on the back end. We believe this will create a much more approachable way to connect schools to multiple local farms, not only reducing food waste, but also providing healthy meals for students and supporting local economies.

Now for the demonstration.

---- CREATE\_DB\_AND\_FILL.py ----

This is the python script that Aimee wrote to create and populate our Database with a mix of real world and generated data. As a note, if it’s not installed already, you will need to run pip3 install psychopg2

---- MPS CSV and MP CSV ----

She found a csv file that contained the needed information of 598 public, K-12 schools, along with a csv that contains data on 532 farms in all 16 counties in Maine.

---- CREATE\_DB\_AND\_FILL.py ----

The generated data we are using is specific produce items that each of the farms would be supplying, purchase orders that functions as transactions between the schools and farms, information about the locations of the hubs which act as way points for food handoffs, and the information about the volunteers that will be responsible for running the hubs.

---- PGADMIN4 – DB ALREADY CREATED ----

All you have to do to create the PostGres database, is clone the git repository and run the Create DB and fill script. now our database has been fully loaded with the information we need for demonstration.

---- FOOD ITEMS TABLE – EXPANDED VIEW ----

The first table we have is the food item table, this table tracks each inventory item that is available to the schools. This table is made up of

* an integer representing A food item’s ID
* A text representation of the food items name
* A text representation of the food item’s description, which is currently NULL but will be filled in by the Farms.
* A text field that represents the macro category of the food item
* An integer representing the available quantity of the food item
* A neumeric representing the cost (in dollars) per unit of the food item
* A date on which the food item was harvested
* A date on which the food item expires
* And lastly, the unit measurement of the food item

---- HUB TABLE – EXPANDED VIEW ----

The next table in our database is for the hub, this table is made up of theoretical hubs that act as meeting points for the schools and farms to reduce logistical burdens. This table is made up of:

* An integer representing each hub’s ID
* A text representation of the hub name
* A text representation of the hub address
* A text representation of the city the hub is in
* A 2 letter VARCHAR representing the state the hub is in
* A 5 digit VARCHAR of the hub zip code
* A text representation of the hub hours
* And finally a text representation of the county the hub is in

---- HUB PRODUCER TABLE – EXPANDED VIEW ----

Next we have our first relationship table, it is our hub producer table, which stores the references for a farm, sometimes referred to as a producer, and it’s associated hub. It is made up of integers for both the hub ID and the producer ID

---- HUB ROLE TABLE – EXPANDED VIEW ----

Our second relationship table is our hub role table. It stores the references for volunteer role shifts and their associated hubs. It is made up of integers for the hub id and the volunteer role ID.

---- HUB SCHOOL TABLE – EXPANDED VIEW ----

Up next is our third relationship table, the hub school table. This table stores the references for the school and it’s associated hub. Both the Hub ID and school ID are integers.

---- PRODUCER TABLE – EXPANDED VIEW ----

Back to the non-relationship tables, we now have the producer table. This table contains all of the information needed for the farms or producers in our project. It consists of:

* An integer for the producer ID
* A text field for the producer name
* A text field for the producers email
* A text field for the producers phone number
* A text field for the producers street address
* A text field for the city or town the producer is located in
* A 2 character VARCHAR for the producers state
* A 5 digit VARCHAR for the producers zip code
* And finally, a text field for the county the producer is in

---- PRODUCER FOOD ITEM TABLE – EXPANDED VIEW ----

Back to the relationship tables, we now have the producer food item table. This table is designed to store the integers for the producer’s ID and their food item ID. This will be useful for when we are creating purchase orders and need to be able to relate a food item being bought to the farm it was supplied by. Or if a hub has a quantity of a particular food item available, but it is supplied by multiple farms. This relation table will be able to ensure the proper farm receives the correct payment.

---- PURCHASE ORDER TABLE – EXPANDED VIEW ----

As previously mentioned, when a school makes a purchase, it will require a purchase order to act as a inventory ticket and a receipt. It consists of:

* An integer for the purchase order ID
* A date for the purchase order transaction date
* An integer representing the unit quantity of an item being purchased
* A numeric of the unit quantities total price
* An integer for the food item’s ID being purchased
* An enum of order status’ for the purchase status. 1 being awaiting fulfillment, 2 being ready at hub and 3 being completed pickup.

---- ROLE SLOT TABLE – EXPANDED VIEW ----

Next up, we have the role slot table. This table is to act as a calendar of sorts to keep track of which shifts and shift roles are possible at the hubs. It consists of:

* An integer for the Volunteer Role ID where each role ID is unique to a shift, position and hub
* A text field for the Role name
* A text field for the role’s description
* A date in year, month, day format for the role needing fulfillment
* And a time without time zone for the role start and role end times.

---- SCHOOL TABLE – EXPANDED VIEW ----

Up next we have the school table where the information for each k-12 public school in Maine is held. In it, we have:

* An integer for the school ID
* A text field for the name of the school
* A text field for the school’s general email address
* A text field for the schools phone number
* A text field for the school’s street address
* A text field for the city or town the school is in
* A 2 character VARCHAR for the school’s state
* A 5 character VARCHAR for the school’s zip code
* And a text field for the county the school is in

---- SCHOOL PURCHASE ORDER – EXPANDED VIEW ----

Our second to last relationship table is the school purchase order table. This table stores the relation between a school and a purchase order it’s made. It contains integers for the school ID and the purchase order ID.

---- VOLUNTEER TABLE – EXPANDED VIEW ----

In the volunteer table, we are storing the information for the members of the community that would like to sign up for shifts at the hubs. The table contains:

* An integer for a volunteer ID
* A text field for a volunteer’s name
* A text field for the volunteer’s email address
* A text field for the volunteer’s phone number
* And an integer for the hub that the volunteer is associated with

---- VOLUNTEER ROLE TABLE – EXPANDED VIEW ----

Our final table is also our last relationship table, the volunteer role table. This table keeps track of the shifts that a volunteer has and will fulfill. It is made up of integers for both the volunteer ID and the volunteer role ID

--------- SQL QUERY WALK THROUGH ---------------

---- EACH QUERY WILL HAVE A READ THROUGH OF THE SQL ----

---- QUERY # 1 – SEARCH ‘Cumberland’ ----

Our first query is to find a hub for a school located in a particular county. At the moment we have one hub per county, but as the database scales, this would be useful for schools to search for all hubs that serve the county they are located in.

---- QUERY #2 SEARCH ‘York’ ----

The second query is finding all producers that are in a particular county. This would allow schools or users to find more information about the Farms that are in their area and offer their contact information.

---- QUERY #3 -- SEARCH ‘Cumberland’ ----

Our third query searches for all schools that are located within a county. This would act as a way for farms to reach out to schools directly to develop a more personal relationship with potential customers.

Check food type Availability by searching a hub's inventory for future availability by macro

---- QUERY #4.a -- SEARCH ‘2’ ----

Utilizing joins and aliases, the first part of our 4th query returns a list of all available food items at a particular hub that is associated with a school.

---- QUERY #4.b -- SEARCH ‘2’ ----

The second part of our 4th works in a similar manner as the first part, but orders the food items by their associated macro category.

These queries join the hub producer table to producer, joins the returned table to producer food item, and joins the returned table to food item. Then it uses where clauses to make sure the returned food items have not expired yet and that they are associated with the desired hub. In 4b. The where clauses also check for the macro attribute.

---- QUERY #5 -- SEARCH ‘2023-04-05 and 2023-05-05’ ----

In our 5th query were trying to find all food items that are expiring within a given date range.

this can be leveraged to highlight or discount food items that need to be moved first to reduce food spoilage and food waste. This query joins the hub producer table to producer, joins the returned table to producer food item, and joins the returned table to food item. Then it uses where clauses to make sure the returned food items expiration date is between the provided date range, and that they are associated with the desired hub.

---- QUERY #6 -- SEARCH ‘eggs’ ----

Our 6th query is intended to allow schools to be able to find producers that have specific food items available. Using aliases and primary keys, this query joins the producer table to producer food item, joins the returned table to food item, and uses a where clause to select only the food item with the specified name.

---- QUERY #7 -- SEARCH ‘379, 2023-04-05 and 2023-05-05’ ----

Our seventh query allows us to return all of a school’s purchase orders for the previous year, or whatever date range is provided. This will be extremely useful for accounting and health analytics for the student body. Using aliases and primary keys, this query joins the school, school purchase order, and purchase order tables. It uses a where clause to find all purchase orders made in a specific date range and returns all the information for those purchase orders.

---- QUERY #8 -- SEARCH ‘48, 2022-08-01 and 2023-08-01’ ----

Query number 8 allows us to search for a producers purchase order history within a given date range. This will be very useful for farm accounting and to track what food items are selling at the highest or lowest rates. This query joins the Purchase order, Food item, producer food item, and producer tables. It returns the purchase order information and uses a where clause to find only the purchase orders connected with a specific farm within a specific date range.

---- QUERY #9.a/9.b -- SEARCH ‘1’ ----

We have broken our ninth query into 2 parts, the first of which will return a list of all possible volunteer positions and shifts at a particular hub sorted by day or time. The second query will return all the open volunteer positions at a particular hub. Both are intended to help volunteers schedule shifts and to see what possible shifts there are in future months. Both these queries join the hub role, role, and hub tables, returning the roles for a specific hub.

**9.b** The second query will also LEFT JOIN the volunteer role table, so when a NULL value is present in the volunteer ID cell, that role hasn’t been filled by a volunteer yet, so we only return the instances where we find NULL values.

---- QUERY #10 -- SEARCH ‘Nick York’ ----

This query is intended for hub volunteers to be able to be contacted by phone. By searching a volunteer’s name their phone number will be returned.

---- QUERY #11 -- SEARCH ‘Cumberland Hub’ ----

This query allows a user to look up a particular hub’s hours of operation by the hub’s name.

---- QUERY #12 -- SEARCH ‘Duff Farms’ ----

This query is intended for producers to be able to be contacted by email. By searching a producer’s name their email address will be returned.

---- QUERY #13 -- SEARCH ‘Deering High School’ ----

This query will return the address of a school when the school’s name is searched.

---- QUERY #14 -- SEARCH ‘7’ ----

This query will show the user all of a farm’s food items and pricing. The query joins the food item, producer food item, and hub producer tables, returning the results ordered by cheapest to most expensive per unit, and also by unit type.

---- QUERY #15 -- SEARCH ‘48, 2022-08-01 and 2023-08-01’ ----

This query will allow a user to see a history of a hub’s purchase orders in a given date range. The query joins the purchase order, school purchase order, school, school hub, and hub tables. It uses a where clause to check for a specific hub, and that the purchase order date is within the specified date range.

---- QUERY #16 -- SEARCH ‘48’ ----

This query will show the user a producer’s orders that are still in progress. The query joins the purchase order, food item, producer food item, and producer tables. It uses a where clause to return only the purchase orders connected with a specific producer AND where the “pur\_status” attribute is a 2, signifying that the purchase order is in progress.

---- QUERY #17 -- SEARCH ‘1’ ----

At a specific hub we’re looking to check what the volunteer role names and their descriptions are, by using aliases and primary keys, this query joins the hub, hub role, and role\_slot tables and returns the role’s name and description for all roles at a specific hub.

---- QUERY #18 -- SEARCH ‘5, 2023-08-05’ ----

This query allows us to identify specific food items at a particular hub that were harvested on a specified date. The query joins the food item, producer food item, hub producer tables, using a where clause to return only the food items connected with a specific hub AND that were harvested between a specified date range.

---- QUERY #19 -- SEARCH ‘1’ ----

To reduce food waste, a user might want to look for items at hubs that have high availability. This query joins the food item, producer food item, and hub producer tables. It groups the food items by name and unit, and returns that food item’s name, unit type, and the sum of the quantity of food item’s with that name at a specific hub.

---- QUERY #20 -- SEARCH ‘5, 50’ ----

If a school is looking to purchase food items that are only available above a certain quantity, this query joins the food item, producer food item, and hub producer tables. It groups the food items by name and unit, and returns that food item’s name, unit type, and the sum of the quantity of food item’s with that name at a specific hub. Although similar to our previous query, it differs because instead of returning all food items grouped this way, it returns only the food items that have a summed quantity greater than the provided amount. Allowing schools to purchase in the quantities needed.

---- QUERY #21 -- SEARCH ‘NEED TO SET EXAMPLE UP’ ----

Using aliases and primary keys, this query joins the purchase order and food item tables, returning all purchase orders that have a specific food item name, food item unit, a purchase order status of 3 , meaning a completed purchase order, and the quantity is above a specified number exclusive of the number input.

---- QUERY #22 -- SEARCH ‘5’ ----

This query will show a user all the schools that are connected to a particular hub which could be used in case of food safety concerns or to gather data analytics on busier hubs.

---- QUERY #23 -- SEARCH ‘2’ ----

This query, much like the previous one, will show a user all the farms/producers that are connected to a particular hub which could be used in case of food safety concerns or to gather data analytics on busier hubs.

---- QUERY #24 -- SEARCH ‘7’ ----

Schools and farms may want to know which food items are frequently being ordered at a hub. This query joins the purchase order, food item, school purchase order, hub school, and hub tables, grouping on food item names, and will return the food item name and the total number of purchase orders with that food item name at a specific hub.

---- QUERY #25 -- SEARCH ‘1’ ----

To reduce food waste, a producer may want to know which items are frequently expiring at hubs either due to lack of interest from schools or difficulty in storage. This query joins the food item, producer food item, hub producer, hub, hub school, and school purchase order tables, grouping by food item name, returning that food item name and a count of distinct food item IDs. This is all done in relation to a specific hub AND makes sure that the expiration date of the food items is before today’s date for historical purposes.

---- QUERY #26 -- SEARCH ’50, 206, 18’ ----

In this query, we reduce the quantity of a food item associated with a specific producer. It takes in the food items ID and quantity to reduce it by, checking to make sure the quantity doesn’t fall below 0. This will be a trigger that happens when a purchase order’s status is changed to a specific status.

---- QUERY #27 -- SEARCH ‘’----

If a farm needs to update its pricing, this query updates the pur total price in the PURCHASE\_ORDER table.