Austin Yuille

Database Systems

Spring 2021

Share Your Crop

Introduction:

New methods of raising crops have appeared as there have been further advancement in agriculture. This includes growing plants in greenhouses, using hydroponics, the introduction of new fertilizers, etc. Also, improvements in technology have made it easier to input and share data across the internet. The appearance of both trends led to me creating a database that can store information about how farms raise and sell their crops to allow for other farmers to look at the crop data of others to hopefully improve the way everyone grows crops.

Topic:

Share your crop is an application that stores information relevant to basic farming practices. For example, you can see all fields and greenhouses on a farm and determine how a farmer irrigates a certain field or if they use hydroponics in some of their greenhouses. You can then further examine the data to see how these trends might affect the crops that these locations produced to see if hydroponics produces crops that yield a higher value at market. The goal behind this application was for farmers to be able to improve the way they grow crops to earn the greatest profit from their labor. This data could also potentially be used to find the most efficient way to grow crops so land can be used more effectively to help the environment.

Scope:

The scope of this project is 20 fake farms and all of their fields, greenhouses, fertilizers, and seeds they might use. It also shows all of the crops they produced and the sales they made with those crops. Disclaimer: Due to the way the data was produced some data may not completely make sense. For example, it is possible that a crop might be harvested before it planted according to the database. This should only occur with dates.

Implementation:

To create this project, I used MySQL to create the schema for the database and to run queries. I created the data using mockaroo.com to produce SQL scripts that I could use to import data into the schemas. For the user interface I used HTML, Python, the Flask library, and Bootstrap. Python, HTML, and Flask were primarily used to create the pages and add functionality to the interface while Bootstrap was used to style the website. I also used Draw.io to create an ER diagram detailing the schema of the database.

ER Diagram:

Diagram

Description automatically generated

Use cases:

The application currently supports adding data to 4 tables: fields, greenhouses, crops, and sales. I chose to add these 4 because for the scope of the project I believed that the fertilizers, seeds, and farms in the database should remain static. Each of the 4 tables has its own page that can be accessed by the “Add Data” drop down in the navigation bar at the top of the screen. Once at an insertion page, the user can enter data for each attribute of the table. All attributes that are foreign keys are entered using drop down list of all possible values to prevent a key being added that does not exist in the referenced table.

The application also allows the user to view any table in the database by using the “View Table” option in the navigation bar at the top of the screen. Once a user begins to view a table they can choose to sort the table by any of the attributes in the table. To do this they simply choose one of the attributes in the drop-down list, and they choose the order for that attribute to be sorted, either smallest to largest or largest to smallest. Once they have selected their sorting preferences they simply press the sort button and the tables will be sorted according to their preferences. This allows the user to search for various trends or group the data in a way that makes it easier for them to find the information they are looking for.

Project Specification Notes:

-Foreign keys used

“farm\_id” (references the farm table) was used in the field and greenhouse tables. “field\_id” (references the field table), “greenhouse\_id” (references the greenhouse table), “seed\_id” (references the seed table), and “fert\_id” (references the fertilizer table) were all used in the crop table. “crop\_id” (references the crop table) was used in the sales table.

-Tables used

farm, field, greenhouse, fertilizer, seed, crop, sales

Conclusion:

Overall, I think the application accomplishes what I wanted it to. Moving forward, I would most likely add joined tables so users could explore more avenues in the data. Also, I would like to have real-world data to use for this application. There were actions that I was prevented from taking due to the way the data was generated, but I believe the current data works fine for proof of concept. I really did learn a lot from this project, especially using Flask and Bootstrap, as these were two tools that I had never used before.

References:

Bootstrap- <https://getbootstrap.com/>

Flask and Bootstrap tutorials- <https://www.youtube.com/watch?v=mqhxxeeTbu0&list=PLzMcBGfZo4-n4vJJybUVV3Un_NFS5EOgX&t=0s>

Mockaroo Data Generator- <https://www.mockaroo.com/>