Dana Davidson

Meghana Shastri

CPSC408

Final Report: Pantry Inventory

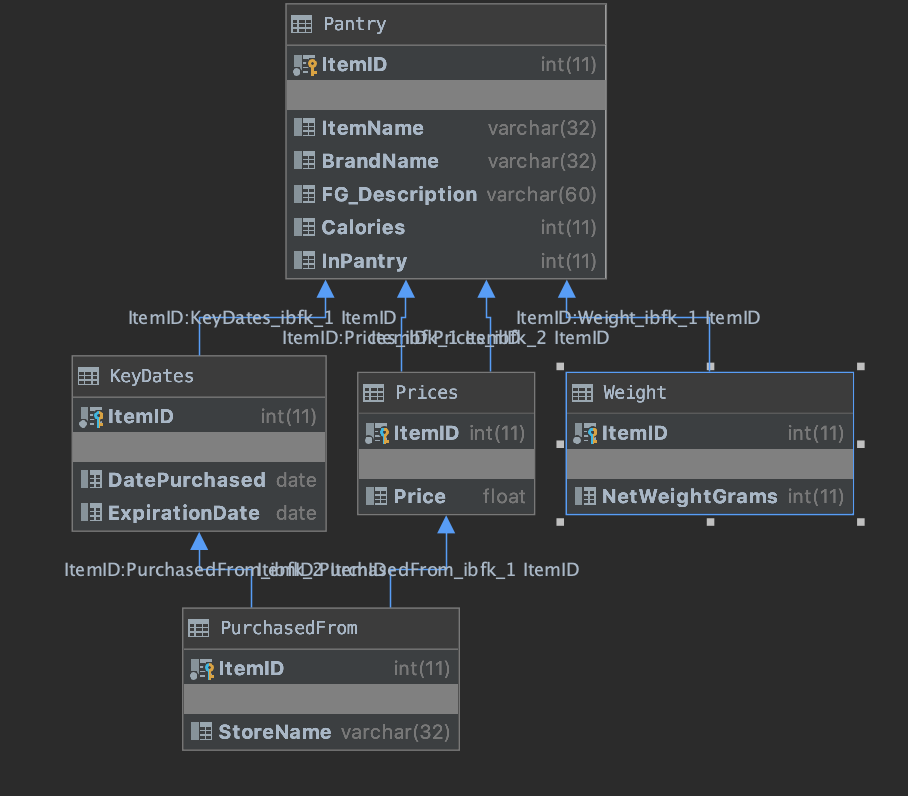
Knowing what ingredients you have on hand can prevent excessive food waste, unnecessary time and money wasted, and can lead to eating out more. When trying to eat in and save money, it’s important to know what food is available in the fridge or pantry. Keeping track of what is still in supply can be a challenge, especially in larger households, where meals are made in larger portions and ingredients can disappear before they’re needed. So our solution was to build an application and database to track the contents of the pantry or fridge as well as other related attributes that may be helpful.

There are many similar apps that are similar to this, but they are often limited in features, usually only including a list of available food and an empty shopping list. These apps have basic add, delete and update, but not much else to aid the user in understanding their spending patterns. We decided to expand on these features, adding attributes like brand, price, date purchased and expiry date to add this kind of functionality.

The pantry inventory application has one listbox using tkinter that displays the user requested records from the database. The buttons along the bottom of the application allow the user to add, update or delete records, as well as see aggregated data like the total sum of all the food in the database. The total cost function is useful for tracking the user’s food spending over time. The user can also export the data to a CSV, see items listed with key dates like when it was added and when it expires and start a search query. The user can search by item name, brand name or any other attribute.

The application still has some kinks we couldn’t work out. For example, the listbox in the GUI does not display all items in the pantry table after adding a new item via the view buttons, but it will display the information in Datagrip. Also, when deleting an item shows in the listbox it will be confirmed as deleted, but the listbox will not show the updated database when clicking view buttons, even though the soft delete works in Datagrip.

For the schema, we had one parent table Pantry connect to three child tables. We did this in order to let the user choose what they want to see in the main table in the listbox. The PurchasedFrom table is a child of KeyDates and Prices in order to create a natural join for two queries, viewing the data with the key dates or viewing the data with the prices of the items.



In all relations, ItemID is the primary key. We used auto\_increment to give an id to every new item to ensure that there are unique items, despite having the same attributes.

In the future it would be interesting to add a connection to a recipe database and recommend recipes based on the ingredients available to the user, or based on the most ingredients that are about to expire to avoid food waste. We also want to make a more user friendly UI. We chose tkinter as the gui but we had trouble along the way, and in retrospect it could have been better to use a web framework such as Django or Flask in order to ease the burden on designing the GUI. We also want to include a table for recipes for the user to reference using the ingredients they have in the pantry.