

Database Project: Hop to It Beer Database

Group:
Thomas Eliassen
Bridger Fisher
Logan Gill
Camden Moore

Date:
12/13/1

ABSTRACT

Hop To It is the premier beer database web application. Written in pure HTML, CSS, and Javascript the minimalist front end gives *Hop To It* a clean, functional look while the Python/Flask backend creates seamless functionality with the MySQL database. Users will be able to expand their love for beer by searching our expansive beer database. *Hop To Its* database features beers from breweries all over the country categorized by style, food pairing, alcohol content and many more. The database also includes breweries from all over the country categorized by region, date established and much more. The largest obstacle *Hop To It* overcame was having an intuitive user interface that integrated with a complex database. The complex functionality came from complex queries sorting and searching the database. In the end *Hop To It* has become a functional prototype that, with further development, can be a fully functional tool for all beer lovers.

Table of Contents

1. Abstract.	i.
2. Problem Description.	1.
3. Database Modeling.	1.
3.1 ER/EER Explanation.	1.
3.2 Database schema explanation.	2.
3.3 SQL.	2.
4. Database App.	3.
5. Summarize.	4.

Project Description

The Hop To It database was designed to be a resource to learn more about beer, breweries, and bars. The website works as a registry where you can look up information about different breweries, learn about their beer and where to find them. The website is easily navigated with the search bar or by clicking buttons and elements. The website is designed to make the database information as accessible and interesting as possible, complete with images, interactive elements, and interesting features. The website was created to help users find new Beers, Breweries, and Bars to expand their interest in beer. The database was created to be robust so that users can execute complex queries and any other functions they desire.

Database Modeling

ER/EER Explanation

Our entities include BEER, BREWERY, BARS, BEER_FOOD, FOOD_PAIRING, BEER_FOR_SALE, BEER_IN_BARS, SALES_LOCATION, BEER_FOR_SALE, BEER_STYLE, REGION, SEASONAL_AVAILABILITY, BEER_FOR_SEASON, PRICE_RANGE

The primary keys from the Beer and Brewery table, are essential to the functionality of the database. The Primary key from the Beer table, Beer_id, is the primary and foreign key in the Bars table, the Food Pairing table, and the Likes table. The primary key from the Brewery table, Brewery_id is a primary and foreign key in the Beer table, the Bars table, and the Likes table. The hierarchy was considered to maximize functionality of our database and our intended use with the web framework.

Database Schema Explanation

Our tables are linked in a strategic manner to create the best use of our database. Many to many connections are avoided by creating linking tables using the primary keys from each side. The cardinality is considered in our database implementation and tested using queries, functions, and testing via our website. We were careful to design the features of the tables to interact well together to avoid any errors.

The best way to describe how our database is structured is to imagine it with one table centered in the middle that works as the core of our database. This is the Beer table which is the center of our database and theme. The Beer database is connected to our breweries, bars and price ranges.

SQL

To create the database, we created many tables including: Beers, Breweries, Bars, Beer_Food, People, Ingredients, and Beer_styles. The data was loaded into these tables by inserting data points into each table after establishing the correct formatting for each tables' features. One of our queries is designed to show the Bars and the Beers that each Bar contains.

Another query we created is designed to show the all of the Breweries. The complex query that we designed shows the Bars and Breweries that they sell beer from.

The Brewery table is connected to the Beers table so that a database user can see which Breweries create the Beers on the database, and so that database users can see the Beers grouped by Brewery. The Bars table is connected to our Beers table so that we can display which Beers database users can find in the various Bars in our database.

Our multi-table query returns the Beers listed by Brewery. In addition another complex query we added is a drop down filter menu on the Brewery page so that users will be

able to see Breweries organized by region. This same functionality can be found on the beers page to filter them by beer style.

Our cursor is somewhat special as it is created from the backend using python. This is one of many great features provided by Python and Flask. It is this cursor that allows us to iterate through rows and display data on our website in seamless manner.

The trigger we created updates the number of beers present in the database for each Brewery so that when the add/remove functionality is utilized the Brewery page will remain up to date with the correct number of beers each brewery has in the database. This decreases database management tasks by streamlining the add/remove process. Our Procedure allows the prices of beers to be changed and will automatically update the Price_Range_Code of the altered Beer. However, this is not a feature we want the public to have access to so it is not accessible from the website. Our View creates a table that displays the Beers organized by Breweries, this too is a functionality that is not useful for our users, but provides functionality when looking at the back end.

Database App

The backend of our app is written in Python with the Flask framework. The tools of our database include our cursor, view, complex and standard queries, our search bar, and the user add/remove functionality. The functions of our database include displaying all beers in the database, displaying all breweries, all bars, and showing the results of a complex query which returns Breweries and a list of their Beers. The user interaction with the database app consists of the home page and header buttons to navigate to the Home, Breweries, Beers, Bars, Mission, and About pages. Each button takes users to a different page with information, and the users can navigate through those pages by clicking on elements. Users can also use the search bar to search for tables.

Our website and database also allows the user to add a new entry, or remove an entry. We have a rough draft of our report, slides, and have continuously improved the website to add user functionality and improve the beautiful design.

Summary

Our team has created a robust database that allows users to learn more about bars, various beers, and breweries around the country. Users can look up beers from different regions and breweries, learn more about the breweries and find out what bars sell their favorite beer. Some of the challenges this project presented was how we made the website express the options available for the user. However, in the end we made a website with icons that represent our intentions. Our functions, simple and complex queries, view, and procedure all work to improve the user experience and provide meaningful information and increase the website functionality.

Future work includes expanding the database content, adding additional beer, breweries, and bars which could be done indefinitely as this content is always changing. With some more work we could start to add beer from other countries. In addition, we would include an administrator page where you would have to login in to be able to add and remove beer. We would also like to include a user system where users can set up profiles to where they can save their favorite beer, bars and leave reviews or organize events. The user system could also have the functionality of interacting with other users and sharing their favorites and creating profiles.