Beer Finder

A database for finding **your** distributor

Our team

Andrew Tate

Just me

Table of Contents

01

Problem Statement

Background Information

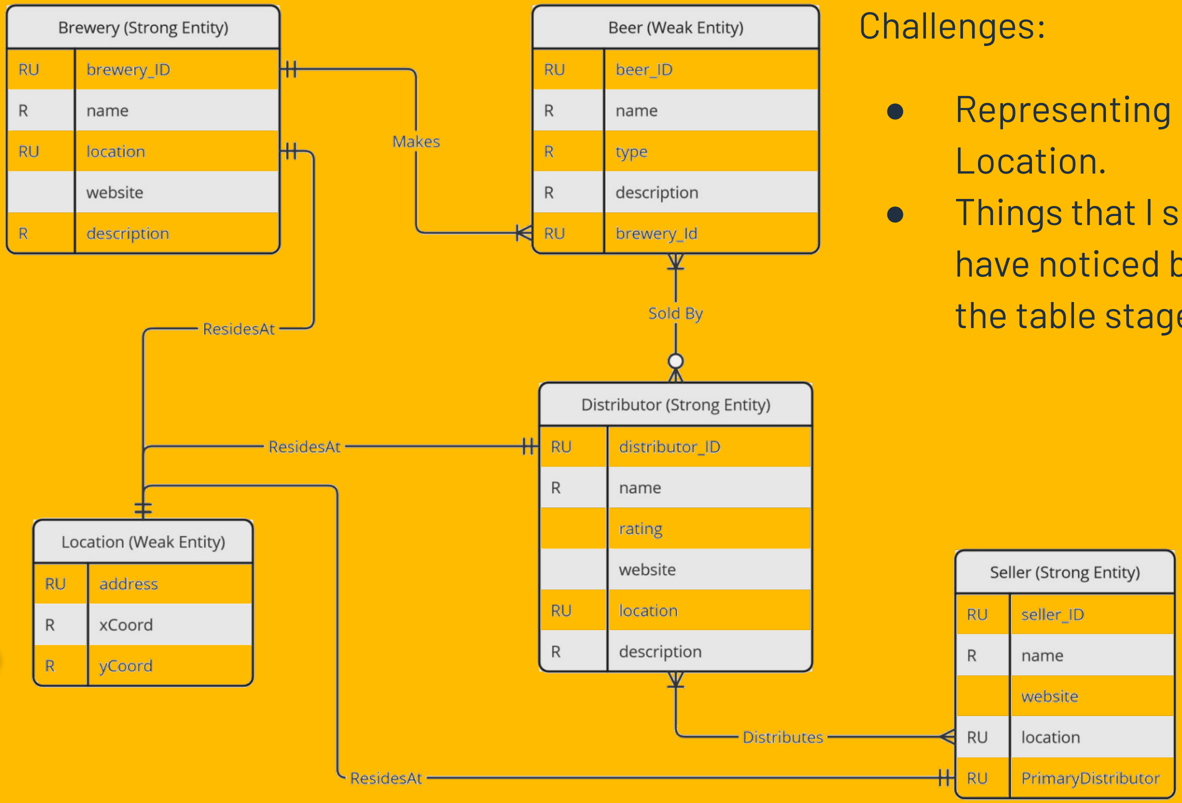
- This database serves to close the gap between Distributors and Licensed Alcohol sellers so that the sellers can more easily pick a distributor that can match their needs
- Database requires locational data, names, descriptions, prices, beer types, inventory for a broad number of entities including breweries, sellers, and distributors
- Came up with this idea from a craft beer shop back home who was having issues with not having a singular place to find this sort of information

Questions the database should be able to answer:

- What are the highest rated distributors near my store's location?
- What is the highest rated beer stocked by *x* distributor?
- Which distributor sells *x* beer for the cheapest price?
- Which distributor sells the most of *x* type of beer?
- Which distributor has the best variety of beer?

02

E-R Diagram

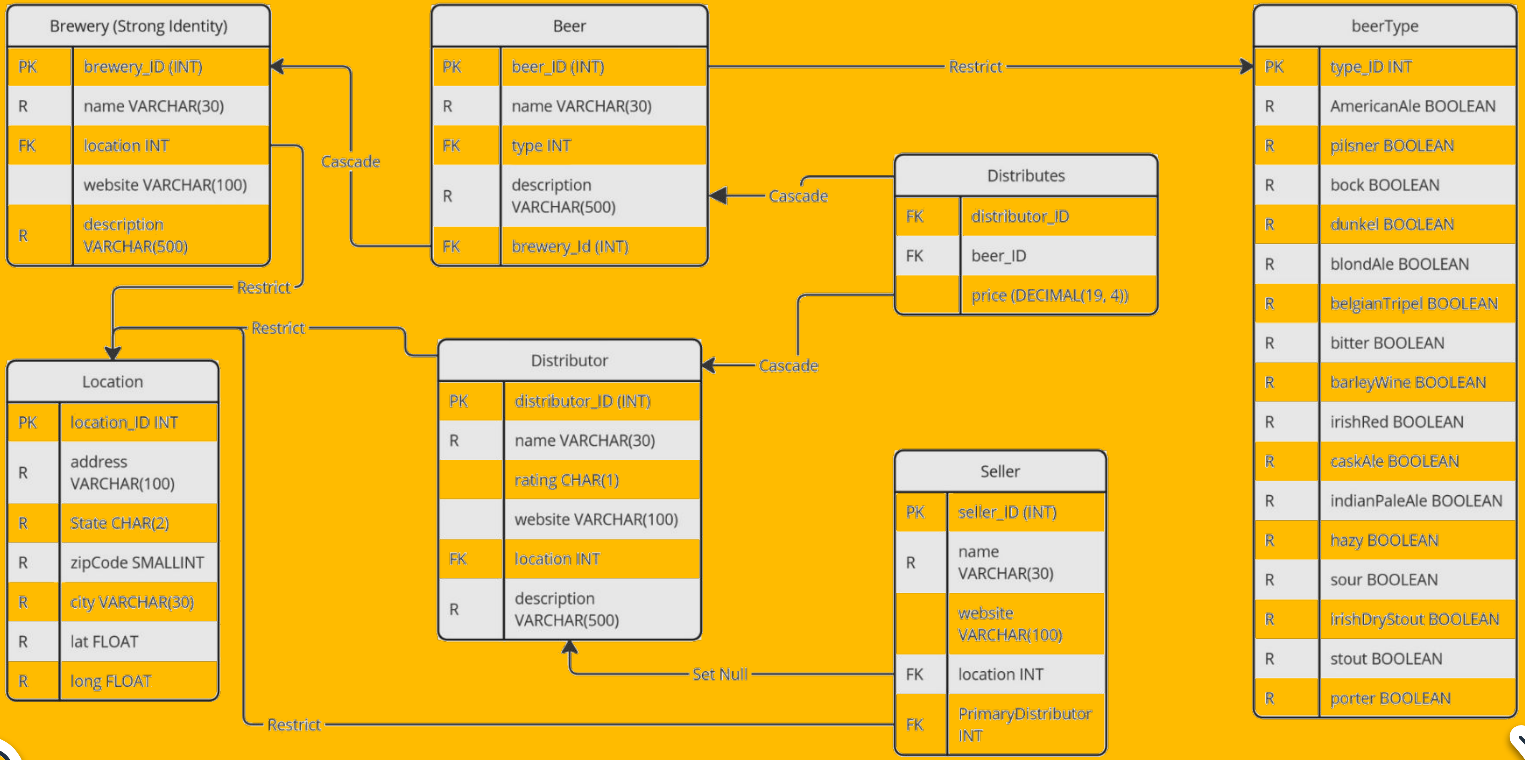


Challenges:

- Representing Location.
- Things that I should have noticed before the table stage.

03

Table Diagram



Brewery (Strong Identity)	
PK	brewery_ID (INT)
R	name VARCHAR(30)
FK	location INT
	website VARCHAR(100)
R	description VARCHAR(500)

Beer	
PK	beer_ID (INT)
R	name VARCHAR(30)
FK	type INT
R	description VARCHAR(500)
FK	brewery_Id (INT)

beerType	
PK	type_ID INT
R	AmericanAle BOOLEAN
R	pilsner BOOLEAN
R	bock BOOLEAN
R	dunkel BOOLEAN
R	blondAle BOOLEAN
R	belgianTripel BOOLEAN
R	bitter BOOLEAN
R	barleyWine BOOLEAN
R	irishRed BOOLEAN
R	caskAle BOOLEAN
R	indianPaleAle BOOLEAN
R	hazy BOOLEAN
R	sour BOOLEAN
R	irishDryStout BOOLEAN
R	stout BOOLEAN
R	porter BOOLEAN

Distributes	
FK	distributor_ID
FK	beer_ID
	price (DECIMAL(19, 4))

Location	
PK	location_ID INT
R	address VARCHAR(100)
R	State CHAR(2)
R	zipCode SMALLINT
R	city VARCHAR(30)
R	lat FLOAT
R	long FLOAT

Distributor	
PK	distributor_ID (INT)
R	name VARCHAR(30)
	rating CHAR(1)
	website VARCHAR(100)
FK	location INT
R	description VARCHAR(500)

Seller	
PK	seller_ID (INT)
R	name VARCHAR(30)
	website VARCHAR(100)
FK	location INT
FK	PrimaryDistributor INT

Notice the horrific beerType table, the many-to-many relationship between beer and distributors, and the expanded location table

Query Examples

```
CREATE OR REPLACE FUNCTION getBestRatedBeerFromDistributor(distributorName
VARCHAR(50), numOfResults INT)
RETURNS TABLE (
    beer_id INT,
    beer_name VARCHAR(50),
    rating CHAR(1),
    beer_description VARCHAR(500)
)
LANGUAGE SQL
AS $$
  SELECT DISTINCT b.beer_id, b.name, b.rating, b.description
  FROM distributor d
  JOIN distributes dis on dis.distributor_id = d.distributor_id
  JOIN beer b on b.beer_id = dis.beer_id
  WHERE d.name = distributorName
  ORDER BY b.rating, b.beer_id
  LIMIT numOfResults;
$$;
```

Notice the differences between mySQL and postgreSQL?

04

CLI Interface

Commands:

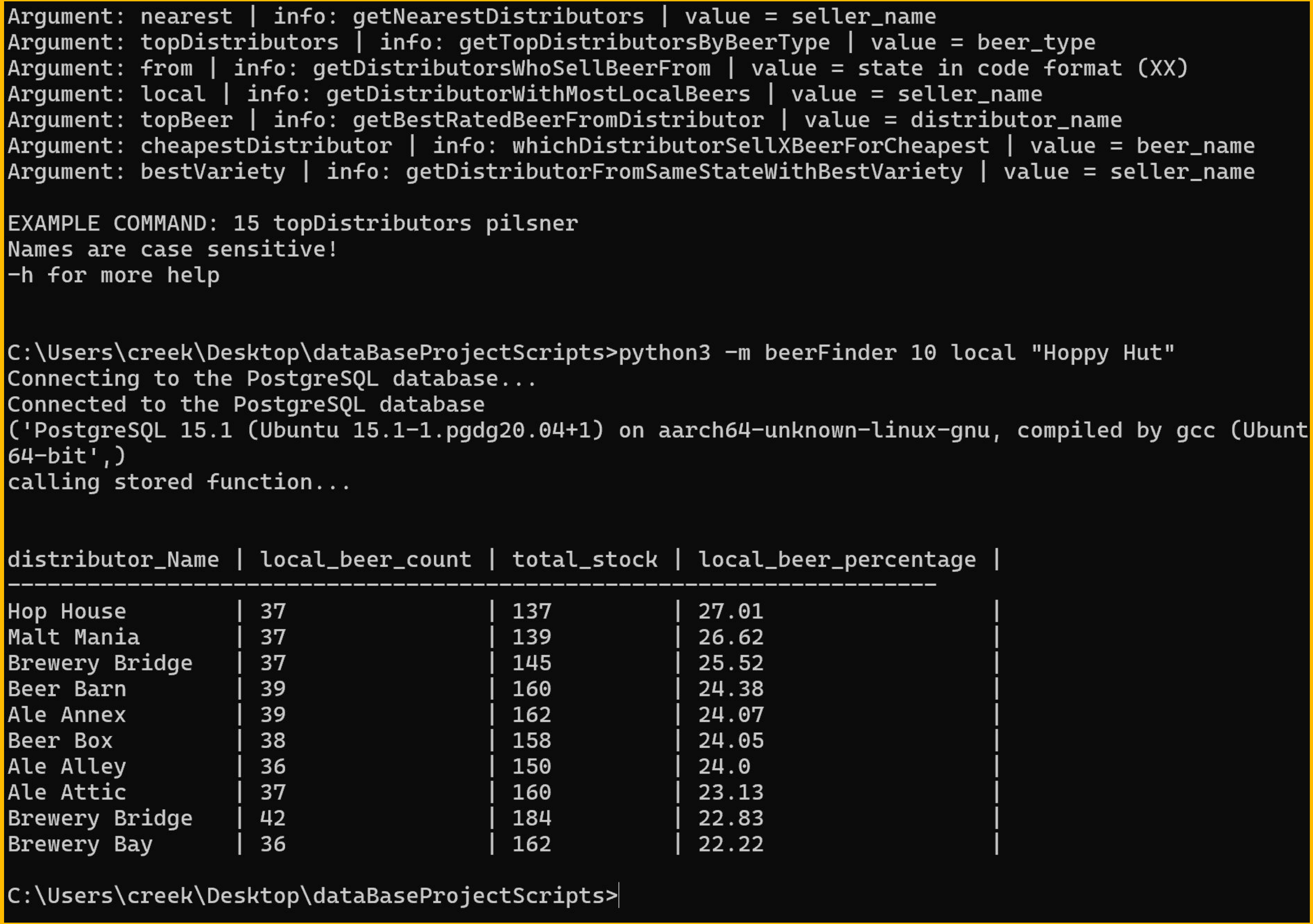
results: an integer representing the number of results you would like to be returned

query: an argument specifying the type of query you would like to do

value: the primary argument associated with the specified query

-l: a list of possible queries with their associated values

-h: help



```
Argument: nearest | info: getNearestDistributors | value = seller_name
Argument: topDistributors | info: getTopDistributorsByBeerType | value = beer_type
Argument: from | info: getDistributorsWhoSellBeerFrom | value = state in code format (XX)
Argument: local | info: getDistributorWithMostLocalBeers | value = seller_name
Argument: topBeer | info: getBestRatedBeerFromDistributor | value = distributor_name
Argument: cheapestDistributor | info: whichDistributorSellXBeerForCheapest | value = beer_name
Argument: bestVariety | info: getDistributorFromSameStateWithBestVariety | value = seller_name

EXAMPLE COMMAND: 15 topDistributors pilsner
Names are case sensitive!
-h for more help


C:\Users\creek\Desktop\dataBaseProjectScripts>python3 -m beerFinder 10 local "Hoppy Hut"
Connecting to the PostgreSQL database...
Connected to the PostgreSQL database
('PostgreSQL 15.1 (Ubuntu 15.1-1.pgdg20.04+1) on aarch64-unknown-linux-gnu, compiled by gcc (Ubunt
64-bit',)
calling stored function...


distributor_Name | local_beer_count | total_stock | local_beer_percentage |
---------------------------------------------------------------------
Hop House        | 37               | 137         | 27.01                 |
Malt Mania       | 37               | 139         | 26.62                 |
Brewery Bridge   | 37               | 145         | 25.52                 |
Beer Barn        | 39               | 160         | 24.38                 |
Ale Annex        | 39               | 162         | 24.07                 |
Beer Box         | 38               | 158         | 24.05                 |
Ale Alley        | 36               | 150         | 24.0                  |
Ale Attic        | 37               | 160         | 23.13                 |
Brewery Bridge   | 42               | 184         | 22.83                 |
Brewery Bay      | 36               | 162         | 22.22                 |

C:\Users\creek\Desktop\dataBaseProjectScripts>
```

05

Limitations/ Implications

- There are around 10,000 breweries in the United States, getting legitimate data for a real version of this application would be a nightmare
- Different countries completely neglect alternative types of beer, so this project might not translate well outside of the United States
- Seasonal Beers mean that data will **constantly** be out of date
- I probably shouldn't store the database's password as a variable in my python file :(

06

Future Work

Ideas:

- Create a full GUI for the application
- Figure out how to create accounts where distributors can only edit table entries associated with their ID and sellers can do the same
- Have some sort of community based data collection
- More queries
- Photos for Breweries, Beer, Sellers, and Distributors
- Travel the country and visit breweries to get a better idea of what data my database might need
- Refactor the beerType table!!!

Thanks!

Remember: You should always drink on days that end in ‘y’

https://github.com/tater-tot25/BeerFinder

Resources

https://www.postgresql.org/docs/current/index.html - Documentation for query syntax

ChatGPT for synthetic data generation

https://www.geeksforgeeks.org/how-to-connect-and-run-sql-queries-to-a-postgresql-database-from-python/ - A good startup guide on how to use psycopg2