**Ryan Fleming**

**CS340 Final Project**

## Outline

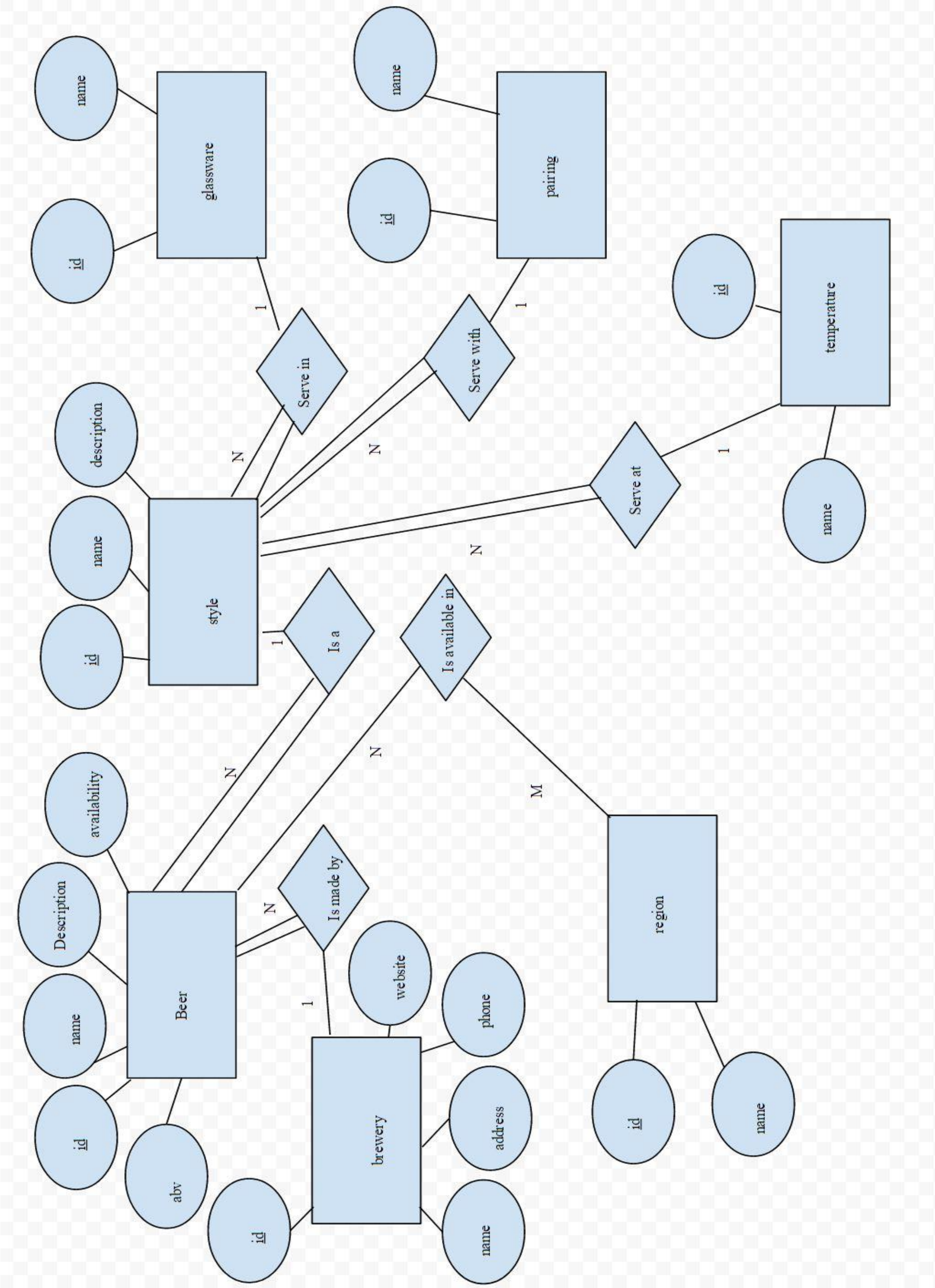
This is a database representing craft beer. Consumers of craft beer are a quickly growing population. To take advantage of this growth, many craft breweries are creating more and more new beers to fill demand as well as new breweries springing up to also take their share of the market. This leads to a lot of new beers that the consumer might not have knowledge about due to a lack of advertising in the industry. Each of these beers has interesting attributes that a consumer might want to search by. Because there is so much information about each beer a consumer might want to know, the complexity needed to describe all of this information makes it a good candidate for a database.

The vast amounts of new products being produced makes it hard to keep track of all of them. Some beers are extremely rare and only available in small quantities in specific regions. This makes it difficult for a small group of people to keep track of and record information on all of them. By having a database that users can add to, it allows more beers to be represented. This way, someone interested in this industry will be able to find information on products that they normally would not be able to.

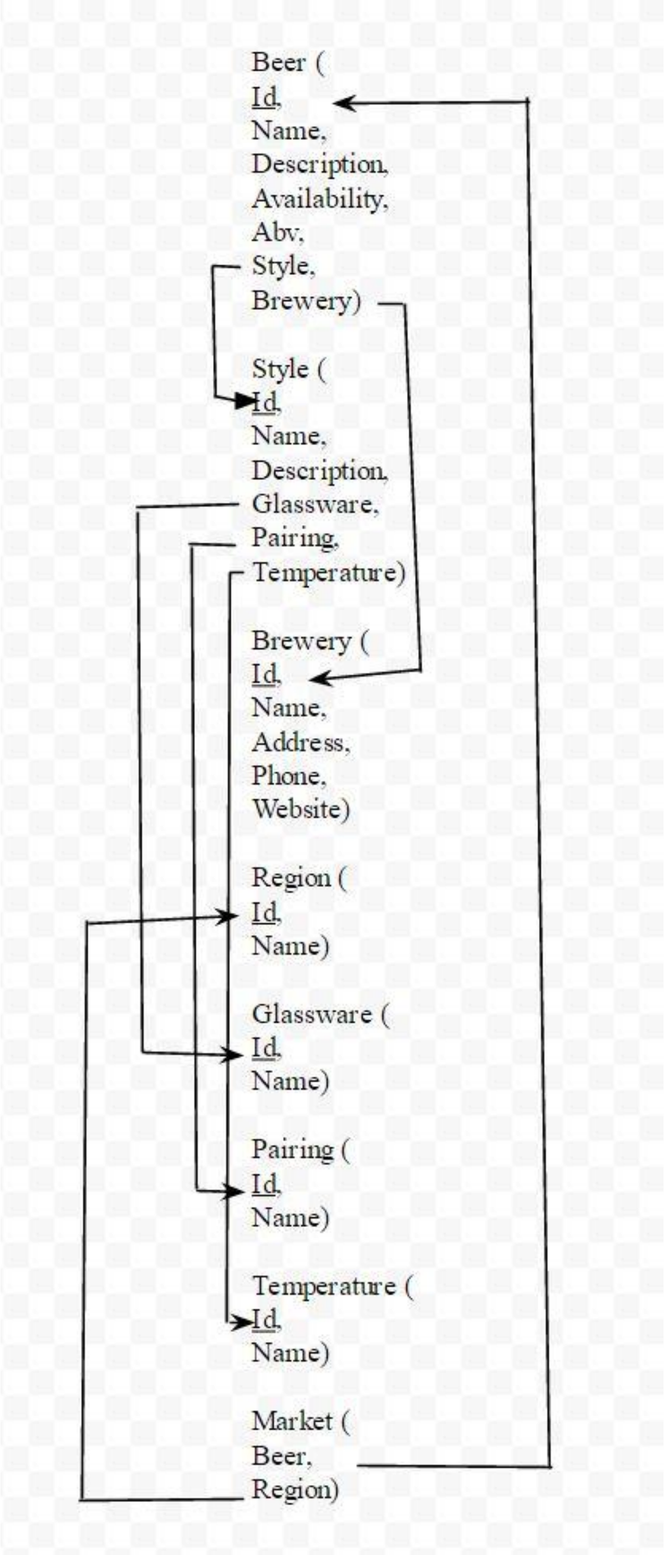
## Database Outline

A beer has an id, name, description, availability, and abv. It is uniquely identified by its id. A style has an id, name, and description. It is uniquely identified by its id. A brewery has an id, name, address, phone and website. It is uniquely identified by its id. A region has an id and name. It is uniquely identified by its id. A glassware has an id and name. It is uniquely identified by its id. A pairing has an id and name. It is uniquely identified by its id. A temperature has an id and name. It is uniquely identified by its id. All ids are auto incrementing. All attributes that are not primary or foreign keys are required. A beer has exactly one style and one brewery. A style has exactly one glassware, one pairing and one temperature. A beer has one or more regions it is available in. A region can have zero or more beers that are available in it. Beer uses a style's id and a brewery's id as foreign keys. Style uses a glassware's id, a paring's id, and a temperature's id as foreign keys. The relationship of beer and regions uses the combination of a beer's id and region's id as a primary key and uses beer's id and region's id as foreign keys. All foreign keys are set to cascade on delete.

## ER Diagram of Database



# Database Schema



Beer (
Id,
Name,
Description,
Availability,
Abv,
Style,
Brewery)

Style (
Id,
Name,
Description,
Glassware,
Pairing,
Temperature)

Brewery (
Id,
Name,
Address,
Phone,
Website)

Region (
Id,
Name)

Glassware (
Id,
Name)

Pairing (
Id,
Name)

Temperature (
Id,
Name)

Market (
Beer,
Region)

## Table Creation Queries

```
CREATE TABLE brewery (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(255) NOT NULL,
address VARCHAR(255) NOT NULL,
phone VARCHAR(255) NOT NULL,
website VARCHAR(255) NOT NULL
) ENGINE=InnoDB;

CREATE TABLE region (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(255) NOT NULL
) ENGINE=InnoDB;

CREATE TABLE glassware (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(255) NOT NULL
) ENGINE=InnoDB;

CREATE TABLE pairing (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(255) NOT NULL
) ENGINE=InnoDB;

CREATE TABLE temperature (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(255) NOT NULL
) ENGINE=InnoDB;
```

```
CREATE TABLE style (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(255) NOT NULL,
description VARCHAR(255) NOT NULL,
glassware_id INT,
pairing_id INT,
temperature_id INT,
FOREIGN KEY (glassware_id) REFERENCES glassware (id) ON DELETE CASCADE,
FOREIGN KEY (pairing_id) REFERENCES pairing (id) ON DELETE CASCADE,
FOREIGN KEY (temperature_id) REFERENCES temperature (id) ON DELETE CASCADE
) ENGINE=InnoDB;

CREATE TABLE beer (
id INT AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(255) NOT NULL,
description VARCHAR(255) NOT NULL,
availability VARCHAR(255) NOT NULL,
abv VARCHAR(255) NOT NULL,
style_id INT,
brewery_id INT,
FOREIGN KEY (style_id) REFERENCES style (id) ON DELETE CASCADE,
FOREIGN KEY (brewery_id) REFERENCES brewery (id) ON DELETE CASCADE
) ENGINE=InnoDB;
```

```
-- relationships

CREATE TABLE market (
beer_id INT NOT NULL,
region_id INT NOT NULL,
PRIMARY KEY (beer_id, region_id),
FOREIGN KEY (beer_id) REFERENCES beer (id) ON DELETE CASCADE,
FOREIGN KEY (region_id) REFERENCES region (id) ON DELETE CASCADE
) ENGINE=InnoDB;

-- insert
INSERT INTO brewery (name, address, phone, website)
VALUES ('Russian River Brewing Company', '725 4th St Santa Rosa, California 95405 United States', '(707)545-2337', 'www.rrbeer.com'), ('The Alchemist', '35 Crossroad Waterbury, Vermont 05676 United States', '(802)244-7744', 'www.alchemistbeer.com'), ('Brouwerij Westvleteren', 'Donkerstraat 12 Westvleteren, 8640 Belgium', '+32 (0)57 40 03 76', 'www.sintsixtus.be'), ('Founders Brewing Company', '235 Grandville Ave SW Grand Rapids, Michigan 49503 United States', '(616)776-1195', 'www.foundersbrewing.com'), ('Brasserie Cantillon', 'Rue Gheude 56 Anderlecht Brussels, 1070 Belgium', '+32 25 21 49 28', 'www.cantillon.be'), ('Goose Island Beer Co.', '1800 W Fulton St Chicago Illinois 60612 United States', '(312)226-1119', 'www.gooseisland.com'), ('3 Floyds Brewing Co.', '9750 Indiana Pkwy Munster, Indiana 46321 United States', '(219)922-4425', 'www.3floyds.com');

INSERT INTO region (name)
VALUES ('Great Lakes'), ('Mid-Atlantic'), ('Midwest'), ('Mountain'), ('New England'), ('Northwest'), ('Pacific'), ('South'), ('South-Atlantic'), ('Southwest'), ('Belgium');

INSERT INTO glassware (name)
VALUES ('Snifter, Tulip, Oversized Wine Glass'), ('Snifter, Tulip, Goblet (or Chalice)'), ('Pint Glass (or Becker, Nonic, Tumbler), Snifter, Oversized Wine Glass'), ('Flute, Snifter, Tulip, Stange (Slender Cylinder)'), ('Flute, Tulip, Oversized Wine Glass'), ('Goblet (or Chalice)'), ('Pint Glass (or Becker, Nonic, Tumbler), Snifter');
```

```
INSERT INTO pairing (name)

VALUES ('Cuisine (Barbecue) Cheese (peppery; Monterey / Pepper Jack, sharp; Blue, Cheddar,
pungent; Gorgonzola, Limburger) Meat (Game, Grilled Meat, Salmon)'), ('Cheese (buttery; Brie,
Gouda, Havarti, Swiss, sharp; Blue, Cheddar) General (Digestive) Meat (Beef, Smoked Meat,
Game)'), ('Cheese (buttery; Brie, Gouda, Havarti, Swiss) General (Chocolate, Digestive) Meat
(Beef, Smoked Meat, Game, Grilled Meat)'), ('General (Chocolate, Salad, Dessert, Apéritif)'),
('Cheese (peppery; Monterey / Pepper Jack, pungent; Gorgonzola, Limburger) General (Salad)'),
('General (Chocolate, Dessert, Digestive)'), ('Cheese (buttery; Brie, Gouda, Havarti, Swiss,
pungent; Gorgonzola, Limburger) General (Chocolate) Meat (Beef)');


INSERT INTO temperature (name)

VALUES ('Cellar @ 45-50F, Serve @ 50-55F'), ('Cellar @ 45-50F, Serve @ 45-50F'), ('Cellar
@ 40-45F, Serve @ 45-50F');
```

## General Use Queries

```
--main page propagation

--select for main table

SELECT beer.name, beer.description, beer.availability, beer.abv, style.name FROM beer
INNER JOIN style ON beer.style_id = style.id


--select for style drop down menu

SELECT id, name FROM style


--select for brewery drop down menu

SELECT id, name FROM brewery


--select for region drop down menu

SELECT id, name FROM region
```

```
--select for glassware drop down menu
SELECT id, name FROM glassware
--select for pairing drop down menu
SELECT id, name FROM pairing

--select for temperature drop down menu
SELECT id, name FROM temperature

--select for beer drop down menu
SELECT id, name FROM beer

--insert for adding into brewery table
INSERT INTO brewery(name, address, phone, website) VALUES
([name],[address],[phone],[website])

--insert for adding into region table
INSERT INTO region(name) VALUES ([name])

--insert for adding into glassware table
INSERT INTO glassware(name) VALUES ([name])

--insert for adding into pairing table
INSERT INTO pairing(name) VALUES ([name])

--insert for adding into temperature table
INSERT INTO temperature(name) VALUES ([name])
```

```
--insert for adding into style table

INSERT INTO style(name, description, glassware_id, pairing_id, temperature_id) VALUES
([name],[description],[glassware_id],[pairing_id],[temperature_id])


-insert for adding into beer table

INSERT INTO beer(name, description, availability, abv, style_id, brewery_id) VALUES
([name],[description],[availability],[abv],[style_id],[brewery_id])


--insert for adding into market table

INSERT INTO market(beer_id, region_id) VALUES ([beer_id],[region_id])


--select for searching beer by name

SELECT beer.name, beer.description, beer.availability, beer.abv, style.name FROM beer
INNER JOIN style ON beer.style_id = style.id WHERE beer.name = [beer.name] GROUP BY
beer.name


--select for sorting by style

SELECT beer.name, beer.description, beer.availability, beer.abv, style.name FROM beer
INNER JOIN style ON beer.style_id = style.id WHERE style.id = [style.id]


--select for sorting by brewery

SELECT beer.name, beer.description, beer.availability, beer.abv, brewery.name FROM beer
INNER JOIN brewery ON beer.brewery_id = brewery.id WHERE brewery.id = [brewery.id]


--select for sorting by region

SELECT beer.name, beer.description, beer.availability, beer.abv, region.name FROM beer
INNER JOIN market ON beer.id = market.beer_id INNER JOIN region on market.region_id =
region.id WHERE region.id = [region.id]
```

```
--select for sorting by glassware

SELECT beer.name, beer.description, beer.availability, beer.abv, glassware.name FROM beer
INNER JOIN style ON beer.style_id = style.id INNER JOIN glassware on glassware.id =
style.glassware_id WHERE glassware.id = [glassware.id]
```

```
--select for sorting by pairing

SELECT beer.name, beer.description, beer.availability, beer.abv, pairing.name FROM beer
INNER JOIN style ON beer.style_id = style.id INNER JOIN pairing on pairing.id =
style.pairing_id WHERE pairing.id = [pairing.id]
```

```
--select for sorting by temperature

SELECT beer.name, beer.description, beer.availability, beer.abv, temperature.name FROM beer
INNER JOIN style ON beer.style_id = style.id INNER JOIN temperature on temperature.id =
style.temperature_id WHERE temperature.id = [temperature.id]
```

```
--delete for deleting a beer from database

DELETE FROM beer WHERE beer.id = [beer.id]
```