Brainstorming

- Users
    - User id
    - Name
    - Region
    - Username
    - Age
    - Profile picture
- Authorization
    - User id foreign
    - Email
    - Password
    - DOB
- Recipes
    - Recipe id
    - public/private
    - User-id (author)
    - Instructions
    - Ingredients list
- Ingredients
    - Ingredient_id
    - Prices
    - Regional availability
    - Recipe id foreign
- Grocery list
    - User id foreign
    - Grocery list id
    - Ingredient id foreign
        - Quantity
    - Estimated price total
- Occasions
    - User id
    - Date
    - Recipe id
    - Occasion id
    - Genre (general occasions)

Table Ideas

- Users
    - Contain user data including IDs and other info that could potentially link them to groups of multiple users with like values
- Authorization
    - Contain login credentials linked to users to grab user id
- Recipes
    - House the recipes themselves including directions as well as lists of ingredients which can link to another ingredients table and can have occasions tags linked to occasions as well. Public/private option, need user id connected to attribute recipe.
- Ingredients
    - This table can simply contain the ingredients needed. May run into complications if there is multiple ways to purchase or prepare an ingredient (garlic, minced garlic…). Link to recipes table to allow for ingredients to be filtered with recipes they are connected to or vice versa. Link to grocery lists to add ingredients and their prices to individual grocery lists.
- Grocery lists
    - I think just lists of ingredients and their prices. Can program to show total estimated price. Link to ingredients. Link to users because each grocery list needs to be unique to a user
- Occasions
    - Link to recipes and to users. Maybe this needs to be 2 tables, like one personal occasions table for users to keep track of their own calendar occasions, and another table to house general occasions like holidays or seasons which can link to recipes. Advanced maybe link to ingredients to do some crunching and determining which ingredients are used most/least at times of the year.
- Personal occasions
    - Link to users and recipes. This could be the table for unique occasions per user, such as birthdays or smaller occasions.

Relationships

- One-one
    - Authorization-user: each authorization will link to one user
- One-many
    - User-recipes: each user can have many recipes and can view other recipes, but each recipe is unique to one user

        - User-grocery lists: each user can have many grocery lists, but those grocery lists may only belong to one user.
        - User-occasions: each user can have multiple occasions, but depending on functionality, there may be unique occasions that only link to one user
- Many-many
    - Recipes-ingredients: many recipes can link with many ingredients
    - Ingredients-grocery lists: many ingredients can link with many grocery lists and vice versa
    - Recipes-occasions: many recipes can link with many occasions and vice versa
    - Users-occasions: depending on how occasions is set up, many users can link with many occasions

Columns

- Users
    - User id -
        - each user needs a unique ID to connect with other tables
        - Integer, primary key because each person needs a unique id
    - Name
        - Human language expression to display users to other humans
        - VARCHAR(30) for first and last name
    - Region
        - I figure if I have time to figure it out it would be neat to allow users to connect and research recipes by region, as well as perhaps regional availability of ingredients(brands/items might be unavailable outside of a particular region)
        - VARCHAR for this since for this exercise I will probably limit it to country, but I was thinking of a potential of subcategories that could be unique and perhaps from a selection like: Continent->country/zone->geographic region. North America/etc->USA/Canada/Mexico->California Coast/Pacific Northwest/Midwest/Southwest/Mountain West/Texas/South/East Coast/Northeast…
    - Username
        - A unique name in case there are multiple people with the same first and last names
        - VARCHAR unique
    - Age
        - People can freely exercise prejudice against age groups in deciding what recipes they like!

        - Integer
    - Profile picture
        - People love visuals and personalization!
        - Text, also selected to accept nulls in case people don't want to upload a picture
- Authorization
    - User id foreign
        - Reference to user id
        - integer
    - Email
        - Need for authorization functions and to verify against bots sorta
        - VARCHAR, unique
    - Password
        - Need for authorization
        - VARCHAR
    - DOB
        - No specific reason yet
        - Date
- Recipes
    - Recipe id
        - Each recipe needs its own unique ID
        - Integer, primary key
    - Recipe name
        - Human language title for recipes to identify
        - VARCHAR
    - public/private
        - If public others can access, but private can keep recipe hidden like a personal recipe book
        - Boolean to simplify functionality
    - User-id (author)
        - To link the user with the recipe that they create
        - Integer, Foreign key
    - Directions
        - Directions are different than ingredients so this is up to the user to explain the process
        - Text - will be very long
    - Ingredients
        - List of ingredients required which will interact with ingredients table. I would like the ingredients to require that they either select an ingredient from the pre-existing list, or if they need to create a new

ingredient it will be added to the ingredients table for future use in recipes. These are another way to connect like-recipes and grocery lists and such.
        - Text. Not entirely sure how to implement what I explained but I figure the simple thing for now is to allow infinite typing of ingredients
- Ingredients
    - Ingredient_id
        - Unique id so that ingredients do not repeat, but can still allow for variations of ingredients with their own unique id's (black beans/canned black beans/Brand black beans)
        - Primary key
    - Recipe id
        - Need to reference recipes
        - Integer, foreign key
    - Prices
        - Would be nice to have a live, updatable list of prices for ingredients so users might be able to estimate their costs and budget meals
        - Float(not sure what is best for a price with 2 decimal places)
    - Regional availability
        - Not sure how to work it, but based on a user's region they would be able to see if the ingredient is unavailable where they live
        - Boolean, perhaps these ingredients can be set to T or F for each world region
- Grocery list
    - User id foreign
        - Grocery lists need to belong to users
        - Integer, foreign key
    - Grocery list id
        - Each grocery list needs its own ID
        - Integer, primary key
    - List name
        - Users can save lists under a name they can refer back to if they want to prepare a recipe again
        - VARCHAR
    - Ingredient id foreign
        - Can pull ingredients and their prices from ingredients table
        - Integer, foreign key
    - Quantity

        - How many of each ingredient needed, can also add price and do calculation
        - VARCHAR, was going to do integer, but then I considered that some items might be bought in unique ways, like 3.5lbs, 2 bunches…
    - Estimated price total
        - Could use the prices and quantities of each item needed to estimate the cost of the entire grocery list
        - Float
- Occasions
    - User id
        - Connect occasions with users
        - Integer, foreign key
    - Date
        - When is the occasion? It could get grouped by month or season for searches
        - Date. Only issue I considered is if it is a variable date occasion (Easter) or if it is an elongated occasion (Lent)
    - Recipe id
        - Connect recipes to occasions or vice versa
        - Integer, foreign key
    - Occasion id
        - Unique occasion id, but I'm at the point where my brain is hurting so I am trying to think about how this might work if it is a public/shared occasion like Christmas - users could select it from the options or create it like the ingredients. Or they could have a private occasion. But I think having a unique id is still necessary so that there aren't a million different occasions all set to Christmas.
        - Integer, primary key
    - Occasion name
        - Another human language representation of the occasion for users to locate.
        - VARCHAR
    - Occasion Type (genre sort of)
        - I envision this as a list of categories that can be chosen in order to group them - Regional Holidays/Religious Holidays/World Holidays/Personal Occasion
        - VARCHAR, but not sure exactly how this would work

```
CREATE TABLE users (
  user_id SERIAL PRIMARY KEY,
  first_name VARCHAR(30),
  last_name VARCHAR(30),
  username VARCHAR(50),
  region VARCHAR(50),
  age INTEGER,
  profile_pic TEXT
  );

CREATE TABLE auth (
  user_id INTEGER REFERENCES users(user_id),
  email VARCHAR(255),
  password VARCHAR(255),
  dob DATE
  );

CREATE TABLE recipes (
  recipe_id SERIAL PRIMARY KEY,
  user_id INTEGER REFERENCES users(user_id),
  public BOOLEAN,
  directions TEXT,
  ingredients TEXT,
  recipe_name VARCHAR(50)
  );

CREATE TABLE ingredients (
  ingredients_id SERIAL PRIMARY KEY,
  recipe_id INTEGER,
  price FLOAT,
  ingredient_name VARCHAR(50),
  regional_availability BOOLEAN
  );

CREATE TABLE grocery_lists (
  grocery_list_id SERIAL PRIMARY KEY,
  user_id INTEGER REFERENCES users(user_id),
  ingredients_id INTEGER REFERENCES ingredients(ingredients_id),
  quantity VARCHAR(255),
  price_est FLOAT,
```

```
  list_title VARCHAR(255)
  );

CREATE TABLE occasions (
  occasion_id SERIAL PRIMARY KEY,
  user_id INTEGER REFERENCES users(user_id),
  recipe_id INTEGER REFERENCES recipes(recipe_id),
  occasion_type VARCHAR(255),
  date DATE
  );
```