Data Modeling Assignment

Part 1: Conceptual Planning

- Step 1: Tracking
  - Recipe Creation
    - Users: name, email, password, id,
    - Recipes: id, name of recipe, who created the recipe, type of recipe(vegan, non-vegan, etc..), serving size, prep time, cook time, public/non-public
    - Instructions: id, tools needed, measurements, ingredients
    - Occasions: id, occasion type
  - Grocery List
    - Users: id, name, email, password, location
    - Recipes: id, items
    - Grocery stores: id, location
- Step 2: Table Ideas
  - Recipe Creation

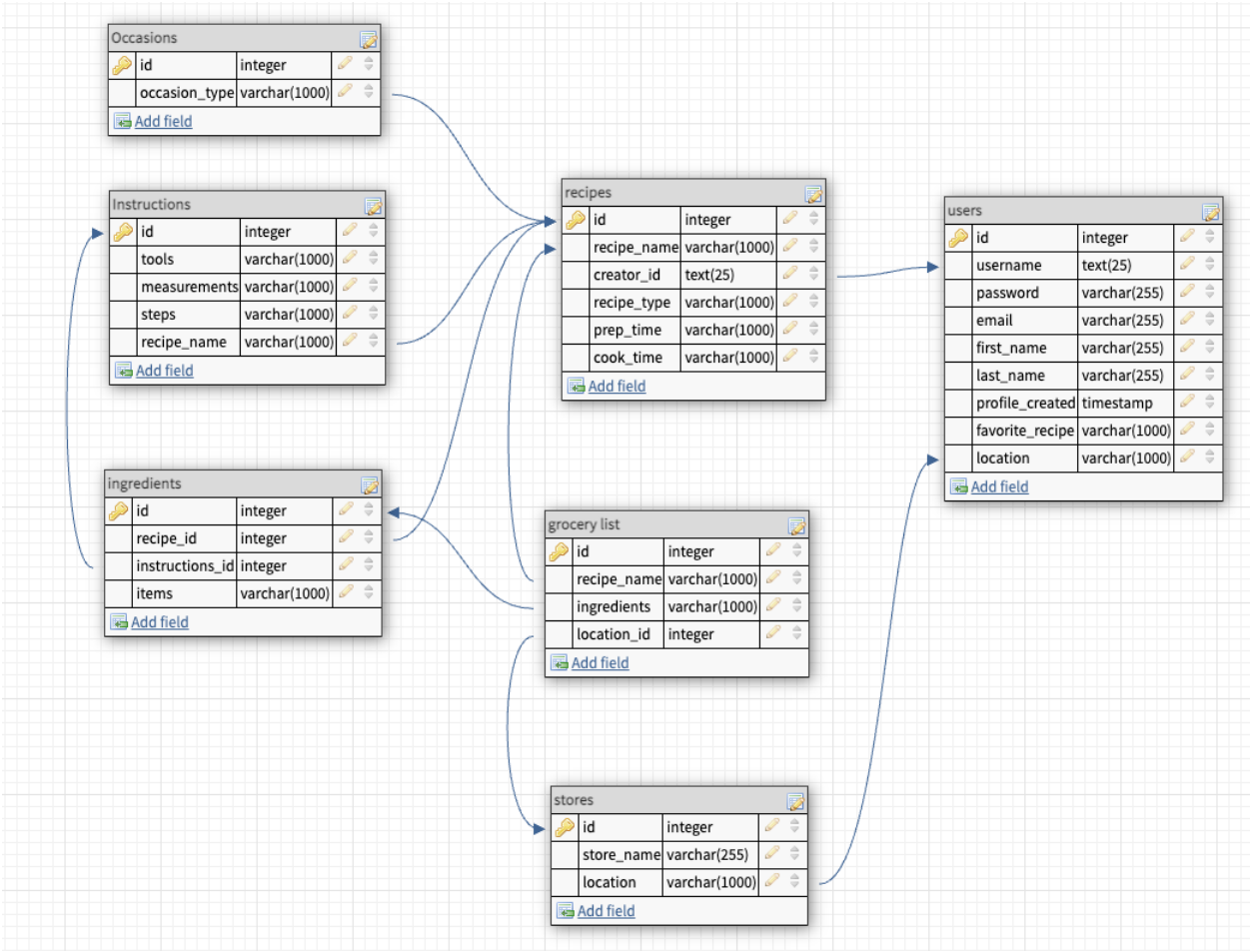
Users - contain information regarding the individual user	Recipes - the table will contain the items needed, this will be the bulk of the data	Instructions - this table will contain all the of tools/measurements needed for the recipes.	Occasions - small table that will contain the type of occasions these meals will be allowed for.	Ingredients - This table will contain the items necessary to create the recipe(s).

  - Grocery List

Users - contain information regarding the individual user, we'll link this table to the recipe users	Recipes - The table will contain the items needed for the recipes	Grocery stores - This table will contain addresses of markets/grocery stores that are in proximity of user(s)

- Step 3: Relationships
  - One-to-one: After reviewing the tables, the one to one table that would correlate would be the recipe and instructions table.
  - One-to-many: After reviewing the tables, there are a few OTM tables that correlate: Users to the Grocery store, users to recipes, recipes to occasions,
  - Many-to-many: Occasions to recipes, instructions to recipes, stores to users
- Step 4: Extra tables
  - Added in one more table specific to only ingredients needed.

Part 2: Table Planning



-
    - Columns:
        - Users
            - Id - Integer
            - Username - Text
            - Password - varchar
            - Email - varchar
            - First name - varchar
            - Last name - varchar
            - Profile created - timestamp
            - Favorite recipe - varchar
            - Location - varchar
        - Recipes
            - Id - integer
            - Recipe name - varchar
            - Creator id - text
            - Recipe type - varchar
            - Prep time - varchar
            - Cook time - varchar
        - Occasions

        - Id - integer
        - Occasion type - varchar
    - Instructions
        - Id - integer
        - Tools - varchar
        - Measurements - varchar
        - Steps - body
        - Recipe_name - varchar
    - Ingredients
        - Id - integer
        - Recipe id - integer
        - Instructions id - integer
        - Items - varchar
    - Grocery list
        - Id - integer
        - Recipe name - varchar
        - Ingredients - varchar
        - Location id - integer
    - Stores
        - Id - integer
        - Store name - varchar
        - Location - varchar
    -
- Tables

Users:

```
CREATE TABLE users(
  id SERIAL PRIMARY KEY,
  username VARCHAR(25),
  password VARCHAR(255),
    email VARCHAR(255),
  first_name VARCHAR(255),
  last_name VARCHAR(255),
  profile_created timestamp,
  favorite_recipe VARCHAR(1000)
  );
```

Recipes:

```
CREATE TABLE recipes(
  id SERIAL PRIMARY KEY,
  recipe_name VARCHAR(1000),
  creator_id VARCHAR(25),
  recipe_type VARCHAR(1000),
  prep_time VARCHAR(1000),
  cook_time VARCHAR(1000)
  );
```

Occasions:

```
CREATE TABLE occasions(
  id SERIAL PRIMARY KEY,
  occasion_type VARCHAR(1000)
 );
```

Instructions:

```
CREATE TABLE instructions(
 id SERIAL PRIMARY KEY,
 tools VARCHAR(1000),
 measurements VARCHAR(100),
 steps VARCHAR(1000),
 recipe_name VARCHAR(1000)
 );
```

Ingredients:

```
CREATE TABLE ingredients(
 id SERIAL PRIMARY KEY,
 recipe_id INTEGER,D
 instructions_id INTEGER,
 steps VARCHAR(1000),
 recipe_name VARCHAR(1000)
 );
```

groceryList:

```
CREATE TABLE groceryList(
 id SERIAL PRIMARY KEY,
 recipe_name VARCHAR(1000),
 ingredients VARCHAR(1000),
 location_id INTEGER
 );
```

Stores:

```
 CREATE TABLE stores(
 id SERIAL PRIMARY KEY,
 store_name VARCHAR(255),
 location VARCHAR(1000)
 );
```