1. List the entity sets and their attributes with primary keys underlined in your design

There are seven entities in our database: Recipes, Ingredients, Recipe Ingredients, Favorites, Tags, Recipe Tags, Grocery List, Inventory, and Nutritional Values. Recipes has eight attributes: recipe_id, recipe_name, instructions, image_url, source_api, last_fetched, is_favorited, and cache_priority. Recipe_id is the primary key and serves as a unique identifier in the API. Recipe_name is a string and holds the name os the recipe. Instructions is a text that contains cooking instructions. Image_url is a string that contains the URL for each recipe image. Source_api is a string that has the API source. Last_fetched is a timestamp stating when the data was last fetched from the API. Is_favorited is a boolean that states whether the recipe is in favorites. Cache_priority is an enum that describes if the recipe is a favorite, recently used, or temporary.

Attributes	Type
recipe_id	Primary Key
recipe_name	String
instructions	Text
image_url	String
source_api	String
last_fetched	Timestamp
is_favorited	Boolean
cache_priority	Enum

The Ingredients entities has three attributes: ingredient_id, ingredient_name, and default_unit. Ingredient_id is the primary key, so it is the unique identifier. Ingredient_name is a string that contains the name of the ingredient. Default_unit is also a string and contains the default measurement unit.

Attribute	Type
ingredient_id	Primary Key

ingredient_name	String
default_unit	String

The Recipe_Ingredients entity has four attributes: recipe_id, ingredient_id, quantity, and unit. Recipe_id is a primary/foreign key and references the recipes. Ingredient_id is a primary/foreign key and references the ingredients table. Quantity is a decimal and just describes the amount needed for that particular recipe. Unit is a string that states the measurement unit for that particular recipe.

Attribute	Type
recipe_id	Primary/Foreign Key
ingredient_id	Primary/Foreign Key
quantity	Decimal
unit	String

The Favorites entity has five attributes: favorite_id, recipe_id, date_added, rating, and notes. Favorite_id is a primary key and the unique identifier for each user's favorite recipe. Recipe_id is a foreign key that references the recipes. Date_added is a timestamp and tracks when the recipe was favorited. Rating is an integer from 1 to 5 that the user gives the recipe, or null if the user has not yet rated the recipe.

Attribute	Type
favorite_id	Primary Key
recipe_id	Foreign Key
date_added	Timestamp
rating	Integer
notes	Text

The Tags entity has three attributes: tag_id, tag_name, and color. Tag_id is a primary key, so it is the unique identifier for Tags. Tag_name is a string that states the tag label. Date_tagged is a timestamp that states when the tag was applied.

Attribute	Type
tag_id	Primary Key
tag_name	String
color	String

The Recipe_Tags has three attributes: tag_id, tag_name, and color. Tag_id is the primary key, so it is the entity's unique identifier. Tag_name is a string that labels the item. Color is also a string which is the hex color code for UI display.

Attribute	Type
recipe_id	Primary/Foreign Key
tag_id	Primary/Foreign Key
date_tagged	Timestamp

The Grocery_List has five attributes: list_id, ingredient_id, quantity, unit, is_purchased. List_id is the primary key, so it is the entity's unique identifier. Ingredient_id is a foreign key that references the ingredients entity. Quantity is a decimal and is the amount to purchase. Unit is a string that is the measurement unit for the item. Is_purchased is a boolean that tracks whether the item has been purchased.

Attribute	Type
list_id	Primary Key
ingredient_id	Foreign Key
quantity	Decimal
unit	String

is_purchased	Boolean

The Inventory has six attributes: inventory_id, ingredient_id, quantity, unit, expiration_date, and date_opened. The inventory_id is the primary key, so it is the unique identifier for each item. The ingredient_id is the foreign key that references ingredients. Quantity is a decimal that indicates the amount of each ingredient in stock. Unit is a string that is the measurement unit used for each ingredient. Expiration_date is a date type that states when the item expires. Date_opened is a date type that indicates when the item was opened.

Attribute	Type
list_id	Primary Key
ingredient_id	Foreign Key
quantity	Decimal
unit	String
is_purchased	Boolean

The Nutritional_Values has nine attributes: nutrition_id, ingredient_id, recipe_id, entity_type, calories, protein, carbs, sugar, and fat. The nutrition_id is the primary key, so it is the unique identifier. Ingredient_id is a foreign key that references ingredients. Recipe_id is a foreign key that references recipes. Entity_type is an enum that states whether the item is an ingredient or a recipe. Calories is a decimal that states the calorie amount in the item. Protein is a decimal that states the protein amount in grams. Carbs is a decimal that states the amount of carbohydrates in grams in the item. Sugar is in decimal which states the amount of sugar in grams in the item. Fat is also in decimal, and it states the amount of fat in grams in the item.

Attribute	Type
inventory_id	Primary Key
ingredient_id	Foreign Key
quantity	Decimal
unit	String
expiration_date	Date

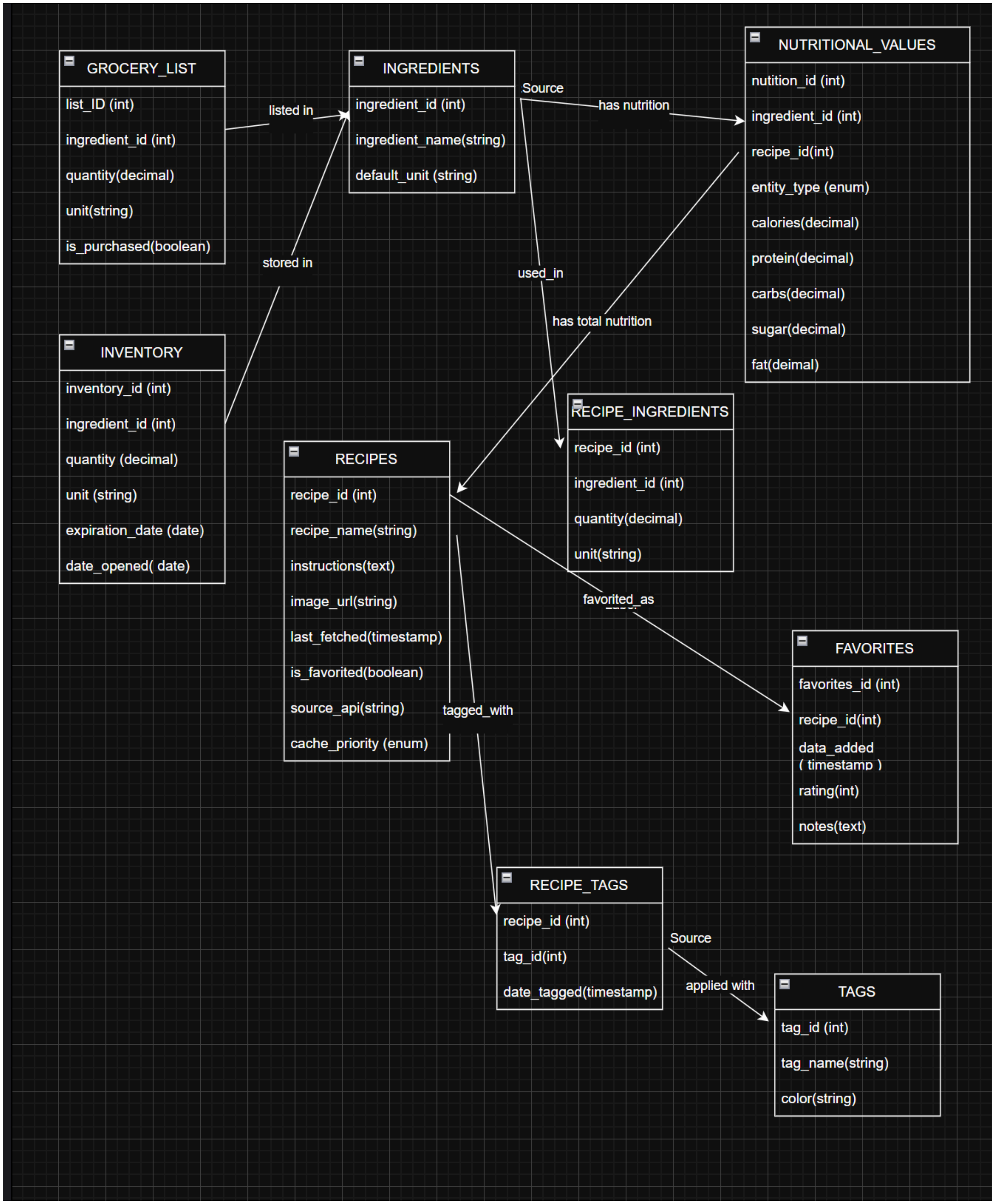
date_opened	Date

2. List the relationships between different entity sets in your design.

There are quite a few relationships among entities. Recipes contain recipe_Ingredients, which has a 1:N cardinality, meaning a recipe can have many ingredients. Ingredients are used in recipe_Ingredients, which have a 1:N cardinality, representing that one ingredient has many recipes. Recipes can be in Favorites, and the relationship has 1:1 cardinality; so each recipe can be favorited only once. Recipes can be tagged with Recipe_Tags, and the cardinality is 1:N, meaning one recipe can have many tags. Tags can be applied to Recipe_Tags, which has a 1:N cardinality, so one tag can be on many recipes. Ingredients can be listed in Grocery_List, which has a 1:N relationship, so one ingredient can appear in a list multiple times. Ingredients can be stored in Inventory and has a cardinality of 1:N, so one ingredient can have multiple inventory entries. Ingredients have Nutritional_Values, which have 1:N cardinality, since each ingredient has nutritional values. Recipes have Nutritional_Values, which is 1:1, since each recipe has total nutrition info.

From Entity	Relationship	To Entity	Cardinality
Recipes	contains	Recipe_Ingredients	1:N
Ingredients	used in	Recipe_Ingredients	1:N
Recipes	can be	Favorites	1:1
Recipes	tagged with	Recipe_Tags	1:N
Tags	applied to	Recipe_Tags	1:N
Ingredients	listed in	Grocery_List	1:N
Ingredients	stored in	Inventory	1:N
Ingredients	has	Nutritional_Values	1:N
Recipes	has	Nutritional_Values	1:1

3.



GROCERY_LIST

list_ID (int)

ingredient_id (int)

quantity(decimal)

unit(string)

is_purchased(boolean)

INGREDIENTS

ingredient_id (int)

ingredient_name(string)

default_unit (string)

NUTRITIONAL_VALUES

nutrition_id (int)

ingredient_id (int)

recipe_id(int)

entity_type (enum)

calories(decimal)

protein(decimal)

carbs(decimal)

sugar(decimal)

fat(deimal)

INVENTORY

inventory_id (int)

ingredient_id (int)

quantity (decimal)

unit (string)

expiration_date (date)

date_opened(date)

RECIPE_INGREDIENTS

recipe_id (int)

ingredient_id (int)

quantity(decimal)

unit(string)

RECIPES

recipe_id (int)

recipe_name(string)

instructions(text)

image_url(string)

last_fetched(timestamp)

is_favorited(boolean)

source_api(string)

cache_priority (enum)

FAVORITES

favorites_id (int)

recipe_id(int)

data_added
(timestamp)

rating(int)

notes(text)

RECIPE_TAGS

recipe_id (int)

tag_id(int)

date_tagged(timestamp)

TAGS

tag_id (int)

tag_name(string)

color(string)

Link:
https://app.diagrams.net/#G1KAZ_aiCrTbrS756fMcrHs9J5LOfSdq4u#%7B%22pageId%22%3A%22YT6NZt9oxexdApxWd8-d%22%7D

4. Provide a detailed explanation of the rationale behind your design choice. Highlight how each element contributes to the overall functionality of your database application.

The rationale behind our design choice is to store recipes efficiently while making it easier for users to store and organize recipes. Recipes are retrieved via an API, and the user can choose to keep them in the favorite entity. Ingredients are stored in a single entity to avoid repetition and to make organizing among recipes, grocery lists, and inventory easier to understand. There are relationships between recipes and ingredients, and between tags to prevent repetition. Favorites, tags, grocery lists, and inventory entities support the users' everyday actions. Nutrition information is also available, which is available for individual ingredients and recipes.