

CPSC 304 Project Cover Page

Milestone #: 2

Date: 21 July 2024

Group Number: 37

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Chris Tang	39077136	s8h2b	christ30625@gmail.com
Mohamed Mohamed	41591876	a6f3h	moemohamed1796@gmail.com
Karen Tran	64835416	r4g2c	karentran28@hotmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

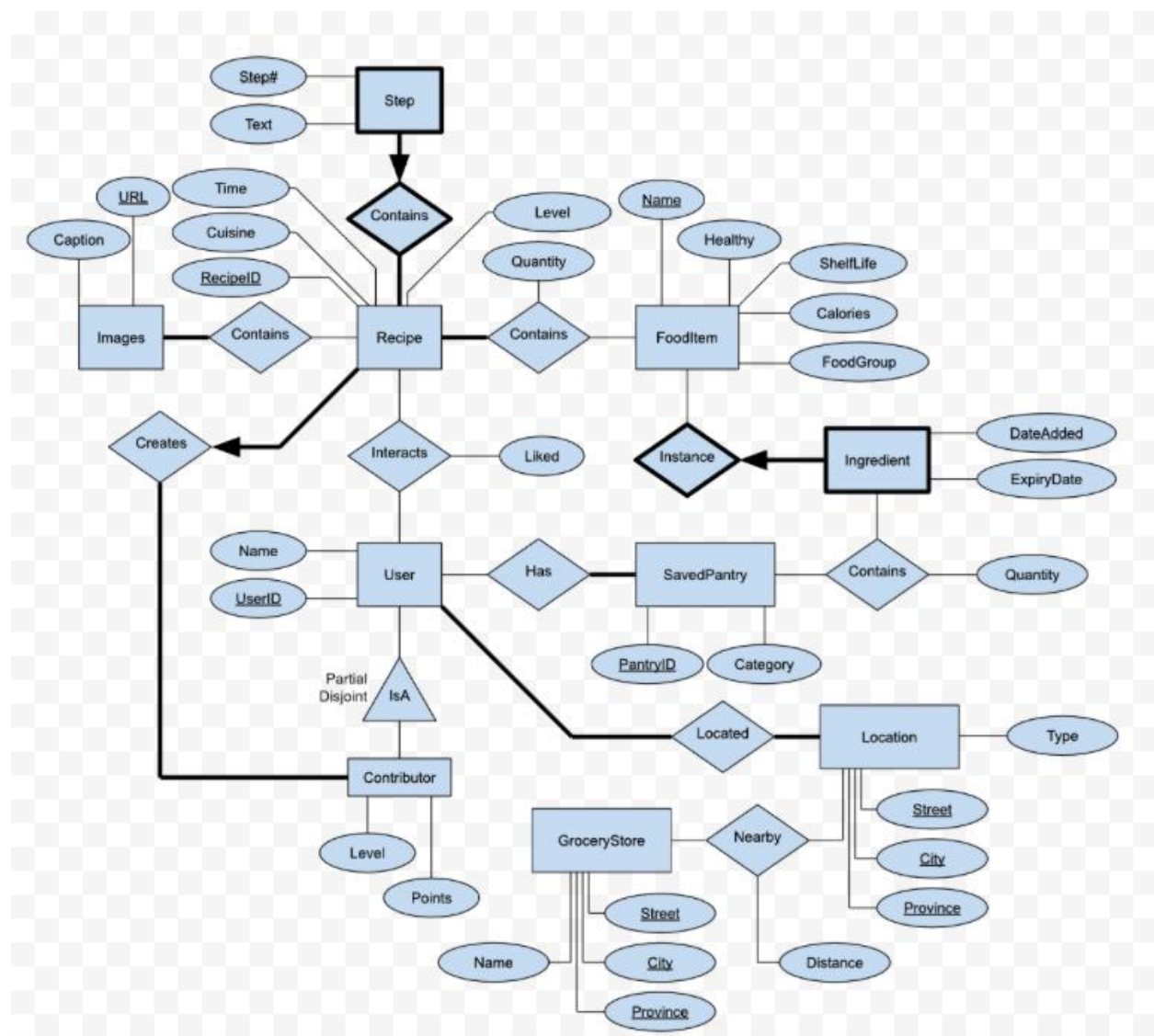
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Milestone 2 - Group 37

Project Summary

Our project is centered around culinary management and grocery planning. The main goal is to help users streamline their cooking activities by keeping track of their household food supplies, identifying exciting new recipes that make the best use of their household inventories, finding nearby grocery stores, and allowing them to contribute recipes towards an online community of like-minded others.

ER Diagram



Several changes were made to our diagram to satisfy the functional dependency requirements. The attribute `Healthy`, a simple boolean value indicating whether the food item is a healthy choice, was added to the `FoodItems` entity. It is functionally dependent on the `Calories` and `FoodGroup` attributes.

The attribute `Level` was added to the `Contributor` subentity. It is an integer value that increases as users contribute new recipes. It is functionally dependent on the `Points` attribute. Related to this, the attribute `Level` was added to the `Recipe` entity and corresponds to the level that must be reached by a contributor before they can access the recipe. It is functionally dependent on the `Cuisine` attribute and will apply to a select number of recipes that we want to restrict to our contributing users.

Schema

Note that there are constraints in our ER diagram that cannot yet be modeled using the material we have been taught in class. For instance, the many-to-many relationship between `User` and `Location` have not (and cannot) be captured using these schema statements. We plan to enforce these constraints at a later date.

PrimaryKey

ForeignKey

PrimaryForeignKey

Constraint

RecipeCreated(<u>RecipeID</u> : int, Cuisine: char(30), Time: time, Level: int, UserID : int (<i>not null</i>), <i>Primary key(s)</i> : RecipeID <i>Foreign key(s)</i> : UserID)
RecipesLiked(<u>RecipeID</u> : int, <u>UserID</u> : int, Liked: tinyint, <i>Primary key(s)</i> : RecipeID, UserID <i>Foreign key(s)</i> : RecipeID, UserID)
Images(<u>URL</u> : varchar(512), Caption: varchar(512), <i>Primary key(s)</i> : URL)
ImagesInRecipes(<u>URL</u> : varchar(512), <u>RecipeID</u> : int, <i>Primary key(s)</i> : URL, RecipeID <i>Foreign key(s)</i> : URL, RecipeID)

FoodsInRecipes(<u>Name</u> : char(30), <u>RecipeID</u> : int, Quantity: int (<i>not null</i>), <i>Primary key(s)</i> : Name, RecipeID <i>Foreign key(s)</i> : Name, RecipeID)
User(<u>UserID</u> : int, Name: char(30), Points: int, Level: int, <i>Primary key(s)</i> : UserID)
UserLocations(<u>UserID</u> : int, <u>Street</u> : char(30), <u>City</u> : char(30), <u>Province</u> : char(30), <i>Primary key(s)</i> : UserID, Street, City, Province <i>Foreign key(s)</i> : UserID, Street, City, Province)
Location(<u>Street</u> : char(30), <u>City</u> : char(30), <u>Province</u> : char(30), Type: char(30), <i>Primary key(s)</i> : Street, City, Province)
NearbyStores(<u>Location.Street</u> : char(30), <u>Location.City</u> : char(30), <u>Location.Province</u> : char(30), <u>GroceryStore.Street</u> : char(30), <u>GroceryStore.City</u> : char(30), <u>GroceryStore.Province</u> : char(30), Distance: float (<i>not null</i>), <i>Primary key(s)</i> : Location.Street, Location.City, Location.Province, GroceryStore.Street, GroceryStore.City, GroceryStore.Province <i>Foreign key(s)</i> : Location.Street, Location.City, Location.Province, GroceryStore.Street, GroceryStore.City, GroceryStore.Province)
GroceryStore(<u>Street</u> : char(30), <u>City</u> : char(30), <u>Province</u> : char(30), Name: char(30) (<i>not null</i>), <i>Primary key(s)</i> : Street, City, Province)

UserPantries(**UserID**: int,
 PantryID: int,
 Primary key(s): UserID, PantryID
 Foreign key(s): UserID, PantryID)

SavedPantry(**PantryID**: int,
 Category: char(30),
 Primary key(s): PantryID)

IngredientsInPantries(**PantryID**: int,
 DateAdded: date,
 Name: char(30),
 Quantity: int (*not null*),
 Primary key(s): PantryID, DateAdded, Name
 Foreign key(s): PantryID, DateAdded, Name)

FoodItem(**Name**: char(30),
 Healthy: tinyint,
 ShelfLife: time,
 Calories: int,
 FoodGroup: char(30),
 Primary key(s): Name)

StepContains(**Step#**: int,
 Text: varchar(512) (*not null*),
 RecipeID: int,
 Primary key(s): Step#, RecipeID
 Foreign key(s): RecipeID)

IngredientInstances(**DateAdded**: date,
 ExpiryDate: date,
 Name: char(30),
 Primary key(s): DateAdded, Name
 Foreign key(s): Name)

Functional Dependencies

RecipeCreated

RecipeID → Cuisine, Time, **UserID**

Cuisine → Level

RecipesLiked

RecipeID, **UserID** → Liked

Images

URL → Caption

ImagesInRecipes

N/A

FoodsInRecipes

Name, **RecipeID** → Quantity

User

UserID → Name, Points

Points → Level

UserLocations

N/A

Location

Street, City, Province → Type

NearbyStores

Location.Street, **Location.City**, **Location.Province**,
GroceryStore.Street, **GroceryStore.City**, **GroceryStore.Province** →
Distance

GroceryStore

Street, City, Province → Name

UserPantry

N/A

SavedPantry

PantryID → Category

IngredientsInPantry

PantryID, **DateAdded**, **Name** → Quantity

FoodItem

Name → ShelfLife, Calories, FoodGroup

Calories, FoodGroup → Healthy

StepContains

Step#, **RecipeID** → Text

IngredientInstances

DateAdded, **Name** → ExpiryDate

Normalization

RecipeCreated

RecipeCreated(RecipeID, Cuisine, Time, **UserID**, Level) →

RecipeCreated₁(Cuisine, Level), RecipeCreated₂(RecipeID, Cuisine, Time, **UserID**)

RecipesLiked

Already in BCNF

RecipesLiked(**RecipeID**: int, **UserID**: int, Liked: tinyint)

Images

Already in BCNF

Images(URL: varchar(512), Caption: varchar(512))

ImagesInRecipes

Already in BCNF

ImagesInRecipes(URL: varchar(512), **RecipeID**: int)

FoodsInRecipes

Already in BCNF

FoodsInRecipes(**Name**: char(30), **RecipeID**: int, Quantity: int (*not null*))

User

User(UserID, Name, Points) →

User₁(Points, Level), User₂(UserID, Name, Points)

UserLocations

Already in BCNF

UserLocations(**UserID**: int, **Street**: char(30), **City**: char(30), **Province**: char(30))

Location

Already in BCNF

Location(**Street**: char(30), **City**: char(30), **Province**: char(30), Type: char(30))

NearbyStores

Already in BCNF

NearbyStores(**Location.Street**: char(30), **Location.City**: char(30), **Location.Province**: char(30), **GroceryStore.Street**: char(30), **GroceryStore.City**: char(30), **GroceryStore.Province**: char(30), Distance: float (*not null*))

GroceryStore

Already in BCNF

GroceryStore(**Street**: char(30), **City**: char(30), **Province**: char(30), Name: char(30))

UserPantries

Already in BCNF

UserPantries(UserID: int, PantryID: int)

SavedPantry

Already in BCNF

SavedPantry(PantryID: int, Category: char(30))

IngredientsInPantries

Already in BCNF

IngredientsInPantries(PantryID: int, DateAdded: date, Name: char(30), Quantity: int (*not null*))

FoodItem

FoodItem(Name, ShelfLife, Calories, FoodGroup) →

FoodItem₁(Calories, FoodGroup, Healthy), FoodItem₂(Name, ShelfLife, Calories, FoodGroup)

StepContains

Already in BCNF

StepContains(Step#: int, Text: varchar(512) , RecipeID: int)

IngredientInstances

Already in BCNF

IngredientInstances(DateAdded: date, ExpiryDate: date, Name: char(30))

SQL DDL Statements

Again, please note that there are constraints in our ER diagram that cannot yet be expressed at this stage. There are plans to enforce them at later stages of the project.

```
CREATE TABLE RecipeCreated1(  
    Cuisine      CHAR(30),  
    Level        INTEGER,  
    PRIMARY KEY (Cuisine)  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE)  
);  
  
CREATE TABLE RecipeCreated2(  
    RecipeID     INTEGER,  
    Cuisine      CHAR(30),  
    Time         TIME,  
    UserID       INTEGER NOT NULL  
    PRIMARY KEY (RecipeID)  
    FOREIGN KEY (UserID) REFERENCES  
        User  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE  
);
```



```

CREATE TABLE RecipesLiked(
    RecipeID    INTEGER,
    UserID      INTEGER,
    Liked       TINYINT,
    PRIMARY KEY (RecipeID, UserID)
    FOREIGN KEY (RecipeID) REFERENCES
        RecipeCreated2
    FOREIGN KEY (UserID) REFERENCES
        User
);

```

```

CREATE TABLE Images(
    URL         VARCHAR(512),
    Caption     VARCHAR(512),
    PRIMARY KEY (URL)
);

```

```

CREATE TABLE ImagesInRecipes(
    URL         VARCHAR(512),
    RecipeID    INTEGER,
    PRIMARY KEY(URL, RecipeID),
    FOREIGN KEY (URL) REFERENCES
        Images
    FOREIGN KEY (RecipeID) REFERENCES
        RecipeCreated2
);

```

```

CREATE TABLE FoodsInRecipes(
    Name        VARCHAR(30),
    RecipeID    INTEGER,
    Quantity    INTEGER NOT NULL,
    PRIMARY KEY (Name, RecipeID).
    FOREIGN KEY (Name) REFERENCES
        FoodItem
    FOREIGN KEY (RecipeID) REFERENCES
        RecipeCreated2
);

```

```

CREATE TABLE User1(
    Points      INTEGER,
    Level       INTEGER,
    PRIMARY KEY (Points)
);

```

```

CREATE TABLE User2(
    UserID      INTEGER,

```

```
        Points          INTEGER,  
        PRIMARY KEY (UserID)  
);
```

```
CREATE TABLE UserLocations(  
    UserID          INTEGER,  
    Street          VARCHAR(30),  
    City            VARCHAR(30),  
    Province        VARCHAR(30),  
    PRIMARY KEY(UserID, Street, City, Province),  
    FOREIGN KEY (UserID) REFERENCES  
        User  
    FOREIGN KEY (Street) REFERENCES  
        Location  
    FOREIGN KEY (City) REFERENCES  
        Location  
    FOREIGN KEY (Province) REFERENCES  
        Location  
);
```

```
CREATE TABLE Location(  
    Street          VARCHAR(30),  
    City            VARCHAR(30),  
    Province        VARCHAR(30),  
    Type            VARCHAR(30),  
    PRIMARY KEY (Street, City, Province)  
);
```

```
CREATE TABLE NearbyStores(  
    Location.Street          VARCHAR(30),  
    Location.City            VARCHAR(30),  
    Location.Province        VARCHAR(30),  
    GroceryStore.Street      VARCHAR(30),  
    GroceryStore.City        VARCHAR(30),  
    GroceryStore.Province    VARCHAR(30),  
    Distance                 INTEGER NOT NULL,  
    PRIMARY KEY(Location.Street, Location.City, Location.Province,  
        GroceryStore.Street, GroceryStore.City,  
        GroceryStore.Province),  
    FOREIGN KEY (Location.Street) REFERENCES  
        Location  
    FOREIGN KEY (Location.City) REFERENCES  
        Location  
    FOREIGN KEY (Location.Province) REFERENCES  
        Location
```

```
        FOREIGN KEY (GroceryStore.Street) REFERENCES
            GroceryStore
        FOREIGN KEY (GroceryStore.City) REFERENCES
            GroceryStore
        FOREIGN KEY (GroceryStore.Province) REFERENCES
            GroceryStore
    );
```

```
CREATE TABLE GroceryStore(
    Street      VARCHAR(30),
    City        VARCHAR(30),
    Province    VARCHAR(30),
    Name        VARCHAR(30) NOT NULL,
    PRIMARY KEY (Street, City, Province)
);
```

```
CREATE TABLE UserPantries(
    UserID      INTEGER,
    PantryID    INTEGER,
    PRIMARY KEY (UserID, PantryID),
    FOREIGN KEY (UserID) REFERENCES
        User
    FOREIGN KEY (PantryID) REFERENCES
        SavedPantry
);
```

```
CREATE TABLE SavedPantry(
    PantryID    INTEGER,
    Category    CHAR(30),
    PRIMARY KEY (PantryID)
);
```

```
CREATE TABLE IngredientsInPantry(
    PantryID    INTEGER,
    DateAdded   DATE,
    Name        VARCHAR(30),
    Quantity    INTEGER NOT NULL,
    PRIMARY KEY (PantryID, DateAdded, Name),
    FOREIGN KEY (PantryID) REFERENCES
        SavedPantry
    FOREIGN KEY (DateAdded, NAME) REFERENCES
        IngredientInstances
);
```

```
CREATE TABLE FoodItem1(
```

```

        Calories    INTEGER,
        FoodGroup   VARCHAR(30),
        Healthy     TINYINT,
        PRIMARY KEY (Calories, FoodGroup)
);

CREATE TABLE FoodItem2(
    Calories    INTEGER,
    FoodGroup   VARCHAR(30),
    Name        VARCHAR(30),
    ShelfLife   TIME,
    PRIMARY KEY (Name)
);

CREATE TABLE StepContains(
    Step#       INTEGER,
    TEXT        VARCHAR(512) NOT NULL,
    RecipeID    INTEGER,
    PRIMARY KEY (Step#, RecipeID),
    FOREIGN KEY (RecipeID) REFERENCES
        RecipeCreated2
        ON DELETE CASCADE
);

```

```

CREATE TABLE IngredientInstances(
    DateAdded   DATE,
    ExpiryDate  DATE,
    Name        CHAR(30),
    PRIMARY KEY (DateAdded, Name),
    FOREIGN KEY (Name) REFERENCES
        FoodItem
        ON DELETE CASCADE
);

```

INSERT Statements

```

RecipeCreated:
INSERT INTO RecipeCreated      VALUES (1, 'Italian', 01:30, 1, 1)
INSERT INTO RecipeCreated      VALUES (2, 'Japanese', 01:15, 2, 2)
INSERT INTO RecipeCreated      VALUES (10, 'Greek', 01:30, 6, 3)
INSERT INTO RecipeCreated      VALUES (28, 'Italian', 00:45, 5, 4)
INSERT INTO RecipeCreated      VALUES (15, 'Italian', 01:00, 3, 5)

```

RecipesLiked:

INSERT INTO RecipesLiked	VALUES (1, 3, 3)
INSERT INTO RecipesLiked	VALUES (19, 5, 30)
INSERT INTO RecipesLiked	VALUES (1, 3, 36)
INSERT INTO RecipesLiked	VALUES (3, 6, 25)
INSERT INTO RecipesLiked	VALUES (8, 5, 43)

Images:

INSERT INTO Images	VALUES('https://xyz.com', 'lasagna')
INSERT INTO Images	VALUES('https://xyz1.com', 'sushi')
INSERT INTO Images	VALUES('https://xyz2.com', 'pizza')
INSERT INTO Images	VALUES('https://xyz3.com', 'ramen')
INSERT INTO Images	VALUES('https://xyz4.com', 'chicken parmesan')

ImagesInRecipes:

INSERT INTO ImagesInRecipes	VALUES('https://xyz4.com', 88)
INSERT INTO ImagesInRecipes	VALUES('https://xyz3.com', 56)
INSERT INTO ImagesInRecipes	VALUES('https://xyz8.com', 34)
INSERT INTO ImagesInRecipes	VALUES('https://xyz9.com', 10)
INSERT INTO ImagesInRecipes	VALUES('https://xyz4.com', 24)

FoodsInRecipes:

INSERT INTO FoodsInRecipes	VALUES('tomato', 1, 2)
INSERT INTO FoodsInRecipes	VALUES('onion', 45, 1)
INSERT INTO FoodsInRecipes	VALUES('garlic', 3, 5)
INSERT INTO FoodsInRecipes	VALUES('bell pepper', 5, 2)
INSERT INTO FoodsInRecipes	VALUES('broccoli', 4, 1)

User:

INSERT INTO User	VALUES(1, 'Karen', 300, 6)
INSERT INTO User	VALUES(2, 'Moe', 87, 9)
INSERT INTO User	VALUES(3, 'Chris', 303, 8)
INSERT INTO User	VALUES(300, 'Bob', 65, 4)
INSERT INTO User	VALUES(499, 'Joe', 0, 1)

UserLocations:

INSERT INTO UserLocations	VALUES(1, '3 University Drive', 'Vancouver', 'British Columbia')
INSERT INTO UserLocations	VALUES(34, '88 Broadway Street', 'Vancouver', 'British Columbia')
INSERT INTO UserLocations	VALUES(67, '77 Fieldstone Way', 'Toronto', 'Ontario')
INSERT INTO UserLocations	VALUES(34, '90 Orange Road', 'Edmontonr', 'Alberta')
INSERT INTO UserLocations	VALUES(99, '8 Blue Willow Drive', 'Vaughan', 'Ontario')

Location:

INSERT INTO Location
'home')

INSERT INTO Location
'work')

INSERT INTO Location

INSERT INTO Location
Columbia', 'home')

INSERT INTO Location

VALUES('8 Blue Willow Drive', 'Vaughan', 'Ontario',

VALUES('90 Yellow Road', 'Vaughan', 'Ontario',

VALUES('8 Orange Way', 'Toronto', 'Ontario', 'home')

VALUES('192 Willow Drive', 'Vancouver', 'British

VALUES('80 Blue Drive', 'Vaughan', 'Ontario', 'work')

NearbyStores:

INSERT INTO NearbyStores

VALUES('80 Blue Drive', 'Vaughan', 'Ontario', '90 Blue
Drive', 'Vaughan', 'Ontario', 2.3)

INSERT INTO NearbyStores

VALUES('3 Yellow Road', 'Toronto', 'Ontario', '10 Blue
Drive', 'Vaughan', 'Ontario', 4.3)

INSERT INTO NearbyStores
Blue

VALUES('230 Brown Drive', 'Vaughan', 'Ontario', '90

Drive', 'Mississauga', 'Ontario', 8.0)

INSERT INTO NearbyStores
Blue

VALUES('90 Salish Drive', 'Vaughan', 'Ontario', '90

Drive', 'Markham', 'Ontario', 2.6)

INSERT INTO NearbyStores
Apple

VALUES('80 Pear Drive', 'Vaughan', 'Ontario', '90

Drive', 'Calgary', 'Alberta', 1.9)

GroceryStore:

INSERT INTO GroceryStore
Mart')

VALUES('80 Pear Drive', 'Vaughan', 'Ontario', 'Sue's

INSERT INTO GroceryStore
'Loblaw')

VALUES('45 Apple Drive', 'Toronto', 'Ontario',

INSERT INTO GroceryStore
'FreshCo')

VALUES('50 Yellow Way', 'Vaughan', 'Ontario',

INSERT INTO GroceryStore

VALUES('77 Marker Drive', 'Vancouver', 'British
Columbia', 'Walmart')

INSERT INTO GroceryStore

VALUES('33 Mouse Lane', 'Edmonton', 'Alberta', 'Farm
Boy')

UserPantries:

INSERT INTO UserPantries

VALUES(1, 2)

INSERT INTO UserPantries

VALUES(4, 27)

INSERT INTO UserPantries

VALUES(8, 28)

INSERT INTO UserPantries

VALUES(5, 20)

INSERT INTO UserPantries

VALUES(10, 8)

SavedPantry:

INSERT INTO SavedPantry	VALUES(2, 'work')
INSERT INTO SavedPantry	VALUES(4, 'home')
INSERT INTO SavedPantry	VALUES(6, 'beach house')
INSERT INTO SavedPantry	VALUES(5, 'airbnb')
INSERT INTO SavedPantry	VALUES(8, 'airbnb')

IngredientsInPantry:

INSERT INTO IngredientsInPantry	VALUES(3, 06-07-2024, 'Broccoli', 1)
INSERT INTO IngredientsInPantry	VALUES(10, 06-08-2024, 'Bread', 4)
INSERT INTO IngredientsInPantry	VALUES(32, 03-09-2024, 'Orange', 6)
INSERT INTO IngredientsInPantry	VALUES(4, 04-09-2024, 'Milk', 3)
INSERT INTO IngredientsInPantry	VALUES(6, 07-20-2024, 'Lemon', 4)

FoodItem:

INSERT INTO FoodItem	VALUES('broccoli', 0, 36h, 200, 'vegetable')
INSERT INTO FoodItem	VALUES('milk', 0, 48h, 250, 'dairy')
INSERT INTO FoodItem	VALUES('orange', 0, 98h, 20, 'fruit')
INSERT INTO FoodItem	VALUES('pasta', 1, 100h, 300, 'grains')
INSERT INTO FoodItem	VALUES('bread', 1, 200h, 450, 'grains')

StepContains:

INSERT INTO StepContains	VALUES(3, 'Preheat oven to 350 deg F')
INSERT INTO StepContains	VALUES(1, 'Chop the onion into cubes')
INSERT INTO StepContains	VALUES(4, 'Place a pan on high heat')
INSERT INTO StepContains	VALUES(2, 'Mince 3 garlic cloves')
INSERT INTO StepContains	VALUES(8, 'Let the dish cook for 30 minutes')

IngredientInstances:

INSERT INTO IngredientInstances	VALUES(07-20-2024, 07-29-2024, 'broccoli')
INSERT INTO IngredientInstances	VALUES(05-20-2024, 06-01-2024, 'milk')
INSERT INTO IngredientInstances	VALUES(06-01-2024, 06-02-2024, 'lemon')
INSERT INTO IngredientInstances	VALUES(04-24-2024, 05-24-2024, 'lime')
INSERT INTO IngredientInstances	VALUES(07-20-2024, 08-30-2024, 'bell pepper')