CPSC 304 Project Cover Page

Milestone #: 2

Date: October 15, 2024

Group Number: 58

Name	Student Number	CS Alias (User ID)	Preferred E-mail Address
Oreoluwa Akinwunmi	10711489	T8j3b	oreakinwunmi@yahoo.com
Hediyeh Mahmoudian	15990880	g3i2f	hediemahmoudian@gmail.com
Helena Sokolovska	37576162	f3e0f	hesoru@gmail.com

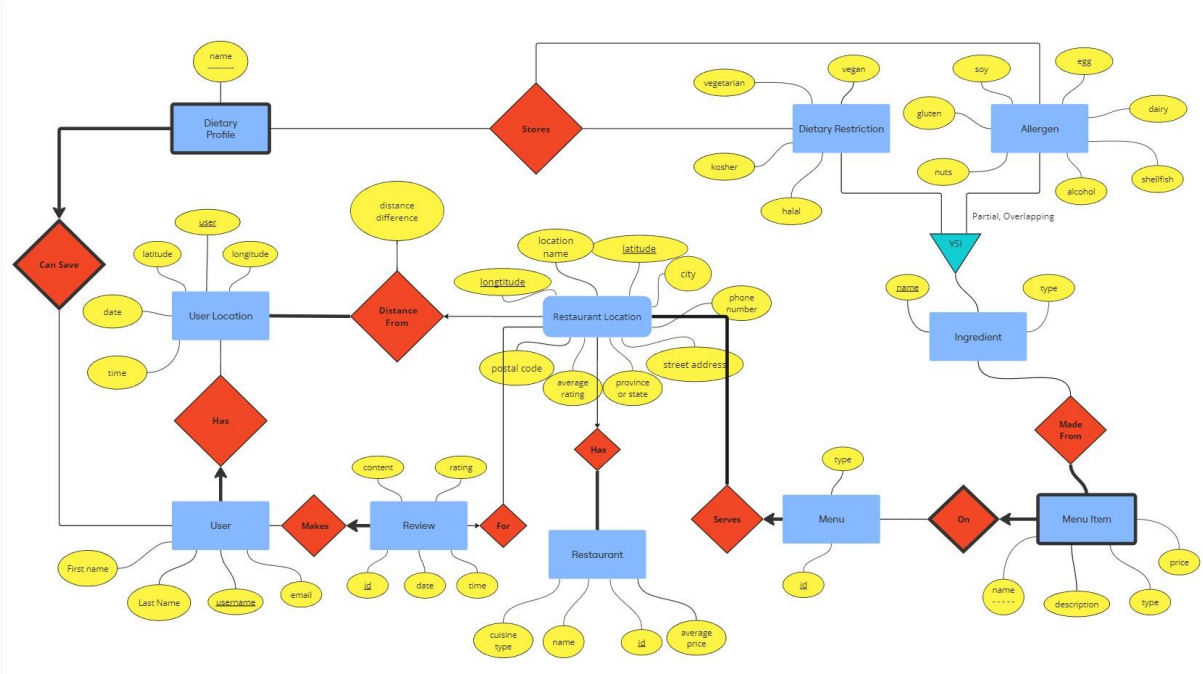
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above.

In 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

(Part 2) Project Summary

This is a database application catered to UBC students to help them navigate all the different restaurants available on campus. It will allow users to search for restaurants nearby based on proximity, menu options, food allergens, dietary restrictions, cuisine type, and affordability.

(Part 3) Updated ER Diagram (larger diagram attached at the end)



Changes

1. **User-User Location Relationship Cardinality**
    a. We changed the **User-User Location** cardinality from a one-to-one relationship to a <u>one-to-many</u> relationship. More specifically, every user MUST have exactly 1 user location, but the same user location can be associated with multiple users (or none at all). This makes more sense since multiple users can be in the same location at once.
2. **User-Location "User" attribute**
    a. We added an attribute (also PK) for **User Location** called <u>user</u> and removed our other 2 PKs (date, time) since changing the cardinality meant the user would identify which location belongs to which person, particularly since 2+ people can have the same location.

3. **User Attributes**
    a. We changed the attributes for **User** from {id, name, email} to {first name, last name, username, email} - this is more consistent with standard user management. We also changed the PK to username as this is the "replacement" for id.
4. **Review Attributes**
    a. To differentiate reviews, we decided to do it by id so we could have a clearer key, in comparison to our previous being date, time and username.
5. **Review-Restaurant & Menu-Restaurant Relationship Changes**
    a. We altered each of these relationships to connect to **Restaurant Location** as we realized reviews and menus will be specific to a *specific restaurant location* rather than a restaurant (chain) as a whole.
6. **Added Attributes to Restaurant Location**
    a. Added *location name, average rating, province/state,city, average rating,* and *phone number*. This provides more detail for our queries.
7. **Added Total Participation Constraint to Restaurant Location**
    a. We added a total participation constraint to restaurant location to say each restaurant location must serve *at least one* menu.
8. **Removed Attributes from Restaurant**
    a. Removed *average rating, low$end, high$end, number of locations* and *phone number*.
9. **Removed/Added item to Menu**
    a. Removed *number of items* and added *type.*

(Part 4) Tables and Keys

(Primary Key: PK, Candidate Key: CK, Foreign Key: FK)

User(*First_Name:* varchar[30], *Last_Name:* varchar[30], *Username:* varchar[30] **(PK),** *Email*: varchar[30] **(CK - NOT NULL, UNIQUE),** *User*: varchar[30] **(FK to User Location)**)

User_Location(*User***:** varchar[30] **(PK)**, *Date:* date, *Time:* time, *Latitude:* decimal(9,6), *Longitude:* decimal(9,6))

Review(*Content:* varchar[200] **(NOT NULL)**, *Rating:* int unsigned **(CHECK (RATING BETWEEN 0 AND 5))**, *Date:* date, *Time:* time, *Id:* unsigned int **(PK)**, *Username*: varchar[30] **(FK to User)**, (*Restaurant_Longitude:* decimal(9,6), *Restaurant_Latitude:* decimal(9,6)) **(FK to Restaurant Location)**)

Restaurant_Location((*Longitude:* decimal(9,6), *Latitude:* decimal(9,6)) **(PK)**, *City:* varchar[30], *Province_or_State:* varchar[30], *Street_Address:* varchar[50] **(CK - NOT NULL, UNIQUE)**, *Postal_Code*: char[6], *Restaurant_Id*: varchar[20] **(FK to Restaurant)**, *Location_Name:* varchar[30], *Phone_Number:* varchar[15] **(UNIQUE),** *Distance_Difference_KM:* decimal(9,6), *Username*: varchar[30] **(FK to User Location),** *Average_Rating:* decimal(3,2) **(CHECK (RATING BETWEEN 0 AND 5))**)

Restaurant(*Id:* varchar[30] **(PK)**, *Cuisine_Type:* varchar[30], *Name:* varchar[30], *Average_Price:* int unsigned)

Menu(*Id:* varchar[30] **(PK),** **Type:* varchar[30], (*Restaurant_Latitude:* decimal(9,6), *Restaurant_Longitude:* decimal(9,6)) **(FK for Restaurant Location)**)
***Menu_Type:** e.g. breakfast, lunch, dinner, dessert, drink, etc.

Menu_Item((*Name:* varchar[30], *Menu_Id:* varchar[30] **(FK for Menu)**) **(PK),** *Description:* varchar[100], **Type:* varchar[30], *Price:* decimal(9,2))
***Menu_Item_Type:** e.g. appetizer, main course, dessert, etc.

Ingredient(*Name:* varchar[30] **(PK),** **Type*: varchar[30])
***Ingredient_Type:** e.g. legume, liquid, oil, etc.

Dietary_Restriction(*Vegetarian:* bool, *Vegan:* bool, *Kosher:* bool, *Halal:* bool, (*Ingredient_Name:* varchar[30] **(FK for Ingredient)** OR (*Dietary_Profile_Name:* varchar[30], *Username*: varchar[30]) **(FK for Dietary Profile)**) **(PK)**)

Allergen(*Nuts:* bool, *Gluten:* bool, *Soy:* bool, *Egg:* bool, *Dairy:* bool, *Shellfish:* bool, *Alcohol*: bool, (*Ingredient_Name:* varchar[30] **(FK for Ingredient)** OR (*Dietary_Profile_Name:* varchar[30], *Username*: varchar[30]) **(FK for Dietary Profile)**) **(PK)**)

Dietary_Profile((*Name:* char[20], *Username*: char[20] **(FK for User)**) **(PK)**)

(Part 5) Functional Dependencies (FDs)

Note: For brevity, (X -> A, B) translates to (X->A) and (X->B). (X & Y -> A) translates to "both X and Y are necessary to determine A." (X | Y -> A) translates to "either X or Y are necessary to determine A."

User:

- (PK) Username -> First_Name, Last_Name, Email
- (CK) Email -> Username, First_Name, Last_Name

User_Location:

- (PK) User -> Latitude, Longitude, date, time

Restaurant:

- (PK) Id -> Average_Price, Cuisine_Type, Name

Restaurant_Location:

- (PK) Longitude & Latitude -> Location_Name, Operating_Hours, Phone_Number, Average_Rating, Street_Address, Postal_Code, City, Province_or_State
- (CK) Street_Address -> Longitude, Latitude, Location_Name, Operating_Hours, Phone_Number, Average_Rating, Postal_Code, City, Province_or_State
- (CK) Phone_Number -> Longitude, Latitude, Location_Name, Operating_Hours, Average_Rating, Street_Address, Postal_Code, City, Province_or_State
- **Postal_Code -> City, Province_or_State**

Review:

- (PK) Id -> Date, Time, Content, Rating

Menu:

- (PK) Menu_Id -> Menu_Type

Menu_Item:

- (PK) Menu_Id & Name -> Description, Type, Price

Ingredient:

- (PK) Name -> Type

Allergen:

- (PK) Ingredient_Name -> Soy, Egg, Dairy, Gluten, Nuts, Shellfish, Alcohol
- (PK) Dietary_Profile_Name & Username -> Soy, Egg, Dairy, Gluten, Nuts, Shellfish, Alcohol

Dietary_Restriction:

- (PK) Ingredient_Name -> Vegan, Vegetarian, Kosher, Halal
- (PK) Dietary_Profile_Name & Username -> Vegan, Vegetarian, Kosher, Halal
- **Vegan -> Vegetarian**

Dietary_Profile:

- (PK) Name, Username

(Part 6) Normalization

1. Restaurant Location:

- (PK) Longitude & Latitude -> Location_Name, Average_Rating, Street_Address, Postal_Code, City, Province_or_State (all attributes of Restaurant_Location)
- (CK) Street_Address -> Longitude, Latitude, Location_Name, Phone_Number, Average_Rating, Postal_Code, City, Province_or_State (all attributes of Restaurant_Location)
- (CK) Phone_Number -> Longitude, Latitude, Location_Name, Average_Rating, Street_Address, Postal_Code, City, Province_or_State (all attributes of Restaurant_Location)
- **Postal_Code -> City, Province_or_State**

A) **Postal_Code** is NOT a superkey, thus **Restaurant_Location** is **not in BCNF**.

B) **R1**(Postal_Code **(PK)**, City, Province_or_State) is in BCNF as **Postal_Code** is a superkey in this relation.

C) **R2**(Longitude, Latitude **(PK)**, Location_Name, Phone_Number, Street_Address Average_Rating, Postal_Code) is in BCNF since all remaining FDs are superkeys.

Rename **R1** to **Postal_Code_Location;** Rename **R2** to **Restaurant_Location.**

New Tables

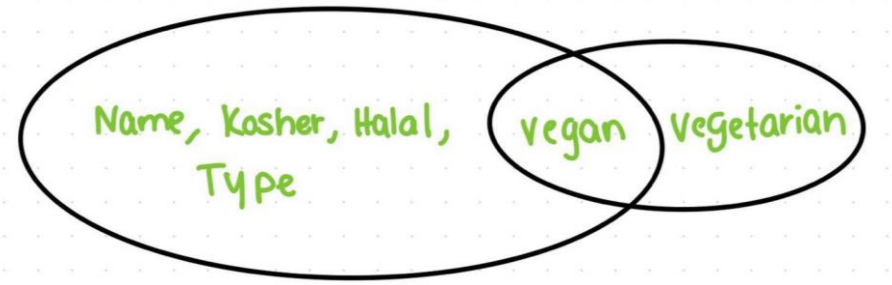
Postal_Code_Location(*Postal_Code:* char[6] **(PK)**, *Province_or_State:* varchar[30], *City:* varchar[30])

Restaurant_Location((*Longitude:* decimal(9,6), *Latitude:* decimal(9,6)) **(PK)**, *Street_Address:* varchar[50] **(CK - NOT NULL, UNIQUE)**, *Restaurant_Id*: varchar[30] **(FK to Restaurant)**, *Postal Code:* char[6] **(FK to Postal_Code_Location)**, *Location_Name:* varchar[30], *Phone_Number:* varchar[15] **(UNIQUE),** **(***Distance_Difference_KM:* decimal(9,6), *Username*: varchar[30]) **(FK to User Location),** *Average_Rating:* decimal(3,2) **(CHECK (RATING BETWEEN 0 AND 5)**)

2. Dietary Restriction:

- (PK) Ingredient_Name -> Vegan, Vegetarian, Kosher, Halal, Type
- **Vegan -> Vegetarian**

A) **Vegan** is NOT a superkey, thus **Dietary_Restriction** is **not in BCNF**.



B) **R1**(Ingredient_Name **(PK)**, Ingredient_Type, Vegan, Kosher, Halal) is in BCNF since Ingredient_Name is a superkey.

C) **R2**(Vegan **(PK)**, Vegetarian) is in BCNF based on the 2-attribute rule.

Rename **R1** to **Dietary_Restriction;** Rename **R2** to **Animal_Product_Restriction.**

New Tables

Dietary_Restriction(*Vegan:* bool **(FK to Animal_Product_Restriction)**, *Kosher:* bool, *Halal:* bool, (*Ingredient_Name:* varchar[30] **(FK for Ingredient)** OR (*Dietary_Profile_Name:* varchar[30], *Username*: varchar[30]) **(FK for Dietary Profile)**) **(PK)**)

Animal_Product_Restriction(*Vegan:* bool **(PK)**, *Vegetarian:* bool)

Remaining Non-Normalized Tables

User(*First_Name:* varchar[30], *Last_Name:* varchar[30], *Username:* varchar[30] **(PK),** *Email*: varchar[30] **(CK - NOT NULL, UNIQUE),** *User*: varchar[30] **(FK to User Location, NOT NULL**))

User_Location(*User***:** varchar[30] **(PK)**, *Date:* date, *Time:* time, *Latitude:* decimal(9,6), *Longitude:* decimal(9,6))

Review(*Content:* varchar[200] **(NOT NULL)**, *Rating:* int unsigned **(CHECK (RATING BETWEEN 0 AND 5))**, *Date:* date, *Time:* time, *Id:* unsigned int **(PK)**, *Username*:

varchar[30] **(FK to User)**, (*Restaurant_Longitude:* decimal(9,6), *Restaurant_Latitude:* decimal(9,6)) **(FK to Restaurant Location)**)

Restaurant(*Id:* varchar[30] **(PK)**, *Cuisine_Type:* varchar[30], *Name:* varchar[30], *Average_Price:* int unsigned)

Menu(*Id:* varchar[30] **(PK),** **Type:* varchar[30], (*Restaurant_Latitude:* decimal(9,6), *Restaurant_Longitude:* decimal(9,6)) **(FK for Restaurant Location)**)
***Menu_Type:** e.g. breakfast, lunch, dinner, dessert, drink, etc.

Menu_Item((*Name:* varchar[30], *Menu_Id:* varchar[30] **(FK for Menu)**) **(PK),** *Description:* varchar[100], **Type:* varchar[30], *Price:* decimal(9,2))
***Menu_Item_Type:** e.g. appetizer, main course, dessert, etc.

Ingredient(*Name:* varchar[30] **(PK),** **Type*: varchar[30])
***Ingredient_Type:** e.g. legume, liquid, oil, etc.

Allergen(*Nuts:* bool, *Gluten:* bool, *Soy:* bool, *Egg:* bool, *Dairy:* bool, *Shellfish:* bool, *Alcohol*: bool, (*Ingredient_Name:* varchar[30] **(FK for Ingredient)** OR (*Dietary_Profile_Name:* varchar[30], *Username*: varchar[30]) **(FK for Dietary Profile)**) **(PK)**)

Dietary_Profile((*Name:* char[20], *Username*: char[20] **(FK for User)**) **(PK)**)

(Part 7 & 8) Creating Tables and Inserting Tuples in SQL DDL

USER

```
CREATE TABLE  User (
        Username VARCHAR(30) PRIMARY KEY,
        First_Name VARCHAR(30),
        Last_Name VARCHAR(30),
        Email VARCHAR(30) NOT NULL UNIQUE
        User VARCHAR(30) NOT NULL
        FOREIGN KEY (User)
                REFERENCES User_Location(User)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

INSERT INTO User
        (Username, First_Name, Last_Name, Email, User)
VALUES
        (hesoru, 'Helena', 'Sokolovska', 'hesoru@gmail.com', '1a'),
```

```
        (hedie, 'Hediyeh', 'Mahmoudian', 'hediemahmoudian@gmail.com', '2b'),
        (oreo, 'Oreoluwa', 'Akinwunmi', 'oreakinwunmi@yahoo.com', '3c'),
        (jsmith, 'Jane', 'Smith', 'jsmith@hotmail.com', '4d'),
        (alovelace, 'Ada', 'Lovelace', 'alovelace@gmail.com', '5e');
```

USER LOCATION

```
CREATE TABLE  User_Location (
        Date DATE,
        Time TIME,
        Latitude DECIMAL(9,6),
        Longtitude DECIMAL(9,6),
        User VARCHAR(30) PRIMARY KEY
);

INSERT INTO User_Location
        (Date, Time, Latitude, Longitude, User)
VALUES
        (15/10/2024, 11:36:14.84000, 49.282730, -123.120735, '1a'),
        (20/12/2020, 15:28:01.39400, 49.267003, -123.248394, '2b'),
        (07/03/2023, 20:59:29.58734, 49.211696, -122.943950, '3c),
        (18/05/2020, 06:25:35.39356, 49.186570, -122.847286, '4d'),
        (01/01/2021, 23:35:55.83490, 49.149756, -123.157886, '5e');
```

REVIEW

```
CREATE TABLE Review (
        Date DATE,
        Time TIME,
        Id INT NOT NULL UNIQUE,
        Rating INT NOT NULL CHECK (Rating BETWEEN 0 AND 5),
        Content VARCHAR(200) NOT NULL,
        Username VARCHAR(30) NOT NULL,
        Restaurant_Latitude DECIMAL(9,6) NOT NULL,
        Restaurant_Longtitude DECIMAL(9,6) NOT NULL,
        PRIMARY KEY Id),
        FOREIGN KEY (Username)
                REFERENCES User(Username)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY (Restaurant_Latitude, Restaurant_Longitude)
                REFERENCES Restaurant_Location(Latitude, Longitude)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);
```

```
INSERT INTO Review
        (Date, Time, Id, Rating, Content, Username, Restaurant_Latitude,
        Restaurant_Longitude)
VALUES
        (13/10/2024, 11:35:14.84070, 1, 4, 'quick and friendly customer service',
        'hesoru', 49.259786, -123.237061),
        (20/09/2023, 15:27:01.39400, 2, 2, 'gross washrooms', 'hedie', 49.262139, -
        123.247368),
        (07/03/2024, 20:34:29.58734, 3, 4, 'the food was hot and fresh', 'oreo',
        49.264491, -123.239292),
        (18/05/2021, 06:11:35.39356, 4, 1, 'don't waste your money!', 'jsmith',
        49.264603, -123.253721),
        (01/01/2020, 23:58:55.83490, 5, 3, 'nothing special, the food was alright',
        'alovelace', 49.268300, -123.243410);
```

RESTAURANT LOCATION

```
CREATE TABLE Restaurant_Location (
        Latitude DECIMAL(9,6),
        Longtitude DECIMAL(9,6),
        Postal_Code CHAR(6) NOT NULL,
        Street_Address VARCHAR(50) NOT NULL UNIQUE,
        Restaurant_Id VARCHAR(30),
        Location_Name VARCHAR(30),
        Phone_Number VARCHAR(15) UNIQUE,
        Username VARCHAR(30),
        Distance_Difference_KM DECIMAL(9,6),
        Average_Rating DECIMAL(3,2) CHECK (Average Rating BETWEEN 0 AND 5),
        PRIMARY KEY (Latitude, Longitude),
        FOREIGN KEY (Restaurant_Id)
                REFERENCES Restaurant(Id)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY (Username)
        REFERENCES User(Username)
                ON DELETE CASCADE
                ON UPDATE CASCADE
        FOREIGN KEY (Postal_Code)
        REFERENCES (Postal_Code_Location)
        ON DELETE CASCADE
        ON UPDATE CASCADE
);

INSERT INTO Restaurant_Location
        (Latitude, Longitude, Postal_Code, Street_Address, Restaurant_Id,
         Location_Name, Phone_Number, Username, Distance_from_User,
         Average_Rating)
```

```
VALUES
        (49.261233, -123.248266, 'V6T1Z4', '6138 Student Union Blvd',
         'R006', 'The Point Grill', '+16048222334', 'oreo', 1.500000, 4.35 ),
        (49.266564, -123.252471, 'V6T1Z4', '2021 West Mall', 'R007',
         'Pacific Poke', '+16048225678', 'maria_smith', 0.800000, 4.20),
        (49.269235, -123.255589, 'V6T1Z4', '6255 Crescent Road', 'R008',
         'Gallery 2 Go', '+16048223456', 'james_lee', 1.200000, 4.00),
         (49.262884, -123.251791, 'V6T1Z4', '2055 Lower Mall', 'R009',
         'Mercante', '+16048224567', 'alovelace', 0.600000, 4.50),
         (49.267726, -123.253154, 'V6T1Z4', '6133 University Blvd', 'R010',
         'Tim Hortons', '+16048226789', 'mike_chan', 0.900000, 3.90);
```

POSTAL CODE LOCATION

```
CREATE TABLE Postal_Code_Location (
        Postal_Code CHAR(6) PRIMARY KEY,
        Province_or_State VARCHAR(30),
        City VARCHAR(30)
);

INSERT INTO Postal_Code_Location
        (Postal_Code, Province_or_State, City)
VALUES
        ('L4C9M3', 'Ontario', 'Richmond Hill'),
        ('V6T1K2', 'British Columbia', 'Vancouver'),
        ('V6S0L4', 'British Columbia', 'Vancouver'),
        ('V6S1Z4', 'British Columbia', 'Vancouver'),
        ('L4E3Z6', 'Ontario', 'Oak Ridges');
```

RESTAURANT

```
CREATE TABLE Restaurant (
        Id VARCHAR(30) PRIMARY KEY,
        Average_Price INT UNSIGNED,
        Cuisine_Type VARCHAR(30),
        Chain_Name VARCHAR(30),
        Restaurant_Latitude DECIMAL(9,6) NOT NULL,
        Restaurant_Longtitude DECIMAL(9,6) NOT NULL,
        FOREIGN KEY (Restaurant_Latitude, Restaurant_Longitude)
                REFERENCES Restaurant_Location(Latitude, Longitude)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);
INSERT INTO Restaurant
        (Id, Average_Price, Cuisine_Type, Chain_Name, Restaurant_Latitude,
         Restaurant_Longitude)
VALUES
        ('R006', 20, 'Grill', 'The Point Grill', 49.261233, -123.248266),
```

```
        ('R007', 15, 'Poke', 'Pacific Poke', 49.266564, -123.252471),
        ('R008', 12, 'Cafe', 'Gallery 2 Go', 49.269235, -123.255589),
        ('R009', 18, 'Italian', 'Mercante', 49.262884, -123.251791),
        ('R010', 10, 'Fast Food', 'Tim Hortons', 49.267726, -123.253154);
```

MENU

```
CREATE TABLE Menu (
        Id INT PRIMARY KEY,
        Type VARCHAR(30),
        Restaurant_Latitude DECIMAL(9,6) NOT NULL,
        Restaurant_Longtitude DECIMAL(9,6) NOT NULL,
        FOREIGN KEY (Restaurant_Latitude, Restaurant_Longitude)
                REFERENCES Restaurant_Location(Latitude, Longitude)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

INSERT INTO Menu
        (Id, Type, Restaurant_Latitude, Restaurant_Longitude)
VALUES
        (1, 'Lunch', 49.261233, -123.248266),
        (2, 'Dinner', 49.266564, -123.252471),
        (3, 'Breakfast', 49.269235, -123.255589),
        (4, 'Pizza', 49.262884, -123.251791),
        (5, 'Coffee', 49.267726, -123.253154);
```

MENU ITEM

```
CREATE TABLE Menu_Item (
Price DECIMAL(9,2) CHECK (Price>0),
        Menu_Id INT,
        Description VARCHAR(100),
        Type VARCHAR(30),
        Name VARCHAR(30),
        PRIMARY KEY (Menu_Id, Name),
        FOREIGN KEY (Menu_Id)
                REFERENCES Menu(Id)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

INSERT INTO Menu_Item
        (Price, Menu_Id, Description, Type, Name)
VALUES
        (12.99, 1, 'Grilled chicken sandwich with fries', 'Lunch', 'Grilled Chicken
         Sandwich'),
        (9.50, 2, 'Salmon poke bowl with rice and veggies', 'Dinner', 'Salmon Poke
```

```
 Bowl'),
        (3.75, 3, 'Fresh brewed coffee, medium size', 'Breakfast', 'Medium Coffee'),
        (15.99, 4, 'Wood-fired Margherita pizza with fresh basil', 'Pizza', 'Margherita
        Pizza'),
        (2.50, 5, 'Small coffee with cream and sugar', 'Coffee', 'Small Coffee');
```

INGREDIENT

```
CREATE TABLE Ingredient (
        Name VARCHAR(30) PRIMARY KEY,
        Type VARCHAR(30)
);

INSERT INTO Ingredient
        (Name, Type)
VALUES
        ('Chicken', 'Meat'),
        ('Rice', 'Grain'),
        ('Broccoli', 'Vegetable'),
        ('Almonds', 'Nut'),
        ('Soy Sauce', 'Condiment');
```

ALLERGEN

(1 = true and 0 = false, defaults to 1 if unspecified)

```
CREATE TABLE Allergen (
        Nuts TINYINT(1) DEFAULT 1,
        Gluten TINYINT(1) DEFAULT 1,
        Soy TINYINT(1) DEFAULT 1,
        Egg TINYINT(1) DEFAULT 1,
        Dairy TINYINT(1) DEFAULT 1,
        Shellfish TINYINT(1) DEFAULT 1,
        Alcohol TINYINT(1) DEFAULT 1,
        Ingredient_Name VARCHAR(30),
        Dietary_Profile_Name VARCHAR(30),
        PRIMARY_KEY(Ingredient_Name, Dietary_Profile_Name),
        FOREIGN KEY (Ingredient_Name)
                REFERENCES Ingredient(Name)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY (Dietary_Profile_Name)
                REFERENCES Dietary_Profile(Name)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

INSERT INTO Allergen
        (Nuts, Gluten, Soy, Egg, Dairy, Shellfish, Alcohol, Ingredient_Name,
```

```
        Dietary_Profile_Name)
VALUES
        (1, 0, 0, 0, 0, 0, 0, 'Almonds', NULL),
        (0, 1, 0, 0, 0, 0, 0, 'Wheat Flour', NULL),
        (0, 0, 1, 0, 0, 0, 0, 'Soy Sauce', NULL),
        (0, 1, 1, 0, 1, 0, 1, NULL, 'Helena's Profile'),
        (0, 0, 0, 0, 1, 0, 0, NULL, 'Jane's Profile');
```

DIETARY RESTRICTION

(1 = true and 0 = false, defaults to 1 if unspecified)

```
CREATE TABLE Dietary_Restriction (
        Halal TINYINT(1) DEFAULT 1,
        Kosher TINYINT(1) DEFAULT 1,
        Vegan TINYINT(1) NOT NULL DEFAULT 1,
        Ingredient_Name VARCHAR(30),
        Dietary_Profile_Name VARCHAR(30),
        Username: VARCHAR(30),
        PRIMARY_KEY(Ingredient_Name, Dietary_Profile_Name),
        FOREIGN KEY (Dietary_Profile_Name, Username)
        REFERENCES (Dietary_Profile)
        ON DELETE CASCADE
        ON UPDATE CASCADE
        FOREIGN KEY (Ingredient_Name)
                REFERENCES Ingredient(Name)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY (Vegan),
                REFERENCES Animal_Product_Restrictions
                ON DELETE CASCADE
                ON UPDATE CASCADE,
);

INSERT INTO Dietary_Restriction
        (Halal, Kosher, Vegan, Ingredient_Name, Dietary_Profile_Name, Username)
VALUES
        (1, 0, 0, NULL, 'Halal', 'hesoru'),
        (0, 1, 0, NULL, 'Sam's Profile', 'samenlox'),
        (0, 0, 1, 'Broccoli', NULL, 'broccolifan400'),
        (0, 0, 1, 'Lentils', NULL, 'hediedoo'),
        (1, 0, 0, 'Lamb', NULL, 'weewoo');
```

ANIMAL_PRODUCT RESTRICTION

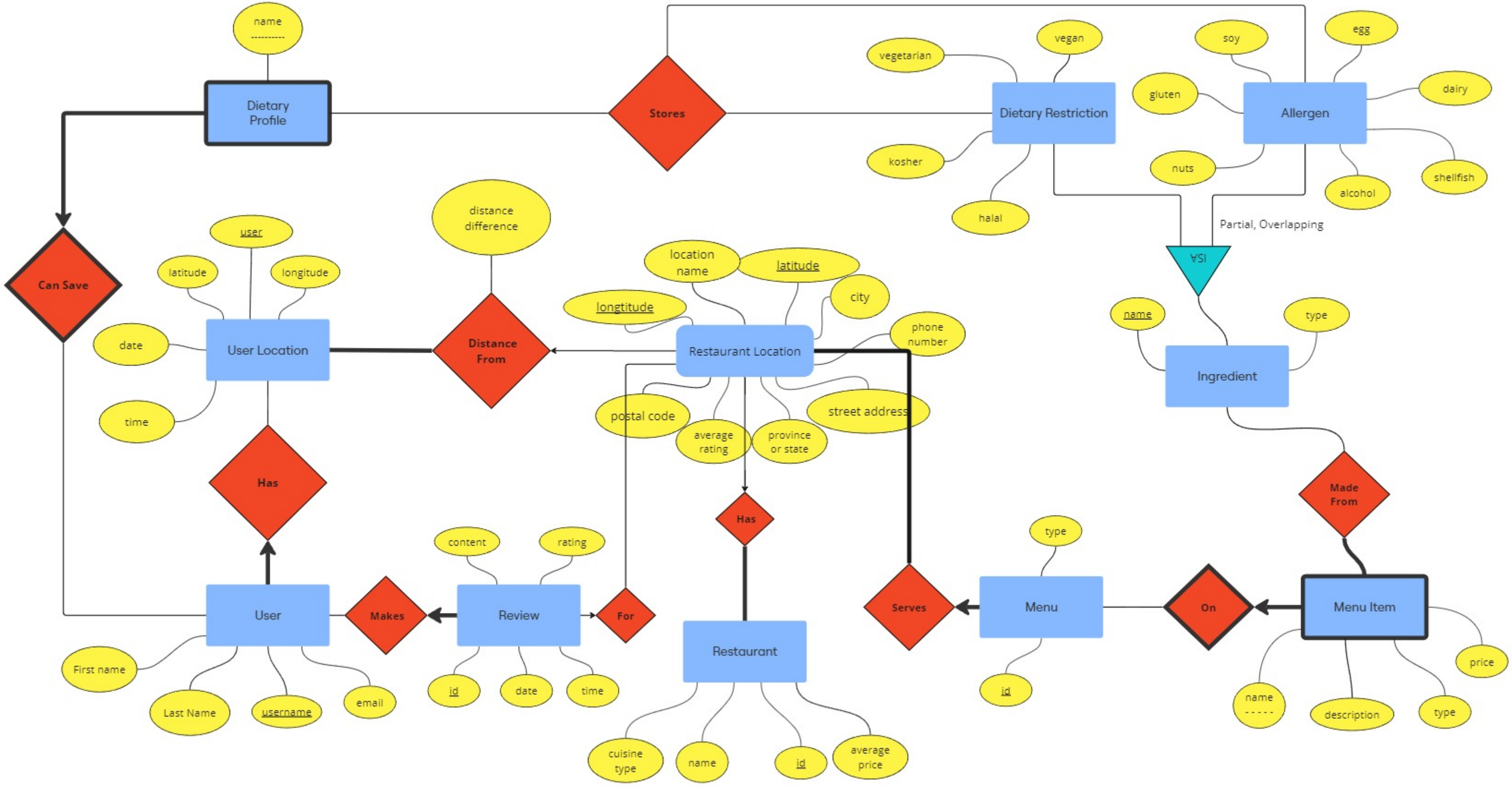
```
CREATE TABLE Animal_Product_Restriction (
        Vegan BOOL PRIMARY KEY,
        Vegetarian BOOL
);
```

```
INSERT INTO Animal_Product_Restriction
        (1, 1),
        (1, 1),
        (0, 1),
        (0, 0),
        (0, 1);
```

DIETARY PROFILE

```
CREATE TABLE Dietary_Profile (
        Name VARCHAR(30) NOT NULL,
        Username VARCHAR(30),
        PRIMARY KEY (Name, Username),
        FOREIGN KEY (Username)
                REFERENCES User(Username)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

INSERT INTO Dietary_Profile
        (Name, Username)
VALUES
        ('Helena's Profile', 'hesoru'),
        ('Vegan', 'jsmith'),
        ('Eating with Dad', 'jlovelace'),
        ('Elimination Diet', 'hedie'),
        ('Halal', 'jsmith');
```


name
Dietary Profile
Stores
vegetarian
vegan
Dietary Restriction
kosher
halal
soy
egg
gluten
Allergen
dairy
nuts
alcohol
shellfish
Partial, Overlapping
ISA
Can Save
user
latitude
longitude
User Location
date
time
distance difference
Distance From
location name
latitude
longtitude
city
phone number
Restaurant Location
postal code
average rating
province or state
street address
name
type
Ingredient
Has
Has
Made From
content
rating
User
Makes
Review
For
Restaurant
Serves
type
Menu
On
Menu Item
First name
Last Name
username
email
id
date
time
cuisine type
name
id
average price
id
name
description
type
price