

CPSC 304 Project Cover Page

Milestone #: 2

Date: October 15 2024

Group Number:

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Alex Luo	17603341	k7i1t	alexluo602@gmail.com
Jerrold Huang	26238998	k5l2e	zhonghan.huang@outlook.com
Jason Liao	67122887	V3d9u	jasonliao999@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 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

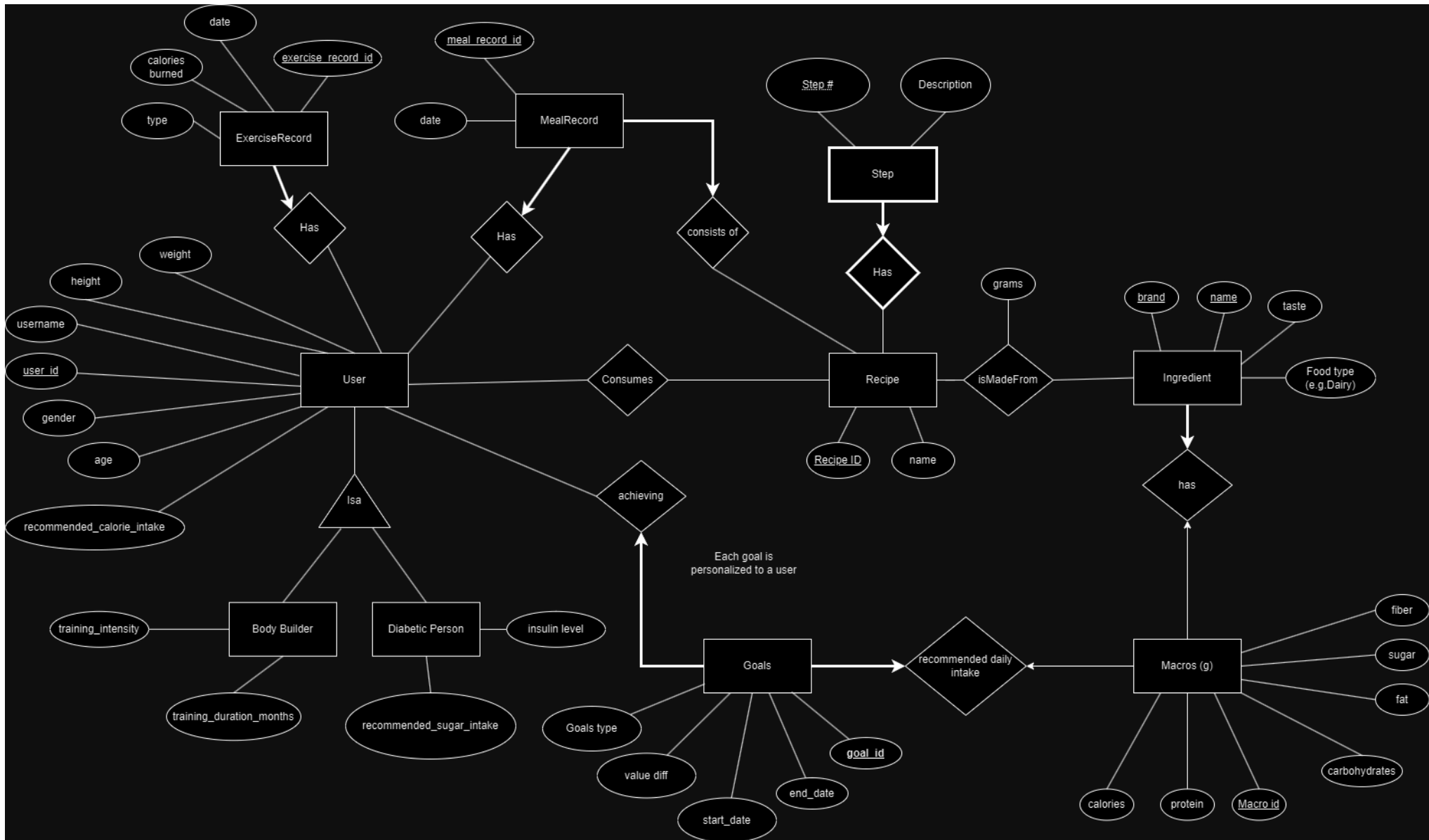
University of British Columbia, Vancouver

Department of Computer Science

2. Brief Summary:

- Personal health management app for daily macro intake logging and monitoring
- Track different goals based on the user's situation (height, weight, etc.)
- Guidance on recipes based on a user's health situation

3. ER Diagram:



Changes to the ER Diagram:

- Renamed **achieves** relationship between user and goals to **achieving** (user is currently working towards the goal)
- We kept total participation on the goals side because we believe that each goal is personalized to a user, and is something the user created.
- We added a **recommended_calorie_intake** attribute to user, which is useful to decide which recipe to make, and also generates more interesting FDs that we can normalize
 - This is just the general recommended calories for a specific body type to maintain current weight and health
 - This differs from recommended daily intake between goals and macros, because that's a recommended macros intake based on the goal you're trying to achieve, which may be different
- We added **training_duration_months** to Body Builder, which helps give more depth to the body builder entity
- We added a **recommended_sugar_intake** to diabetic person to help any diabetic person in deciding what meal they want to make based on their sugar levels.

4. Relational Models

Macros (macro_id: INT, fiber: INT, sugar: INT, fat: INT, carbohydrates: INT, protein: INT, calories: INT)

Primary Key: macro_id
Candidate Key: macro_id
ForeignKey:

Ingredient (name: VARCHAR(255), brand: VARCHAR(50), taste: VARCHAR(255), food_type: VARCHAR(50), macro_id: INT);

Primary Key: (name, brand)
Candidate Key: (name, brand)
ForeignKey: macro_id references Macros(macro_id)
Constraints: macro_id NOT NULL and UNIQUE

University of British Columbia, Vancouver

Department of Computer Science

MealRecord (meal_record_id: INT, date: DATE, user_id: INT, recipe_id: INT)

Primary Key: meal_record_id
Candidate Key: meal_record_id
ForeignKey: **user_id** references User(user_id), **recipe_id** references Recipe(recipe_id)
Constraints: user_id NOT NULL, recipe_id NOT NULL

Goals(goal_id: INT, start_date: DATE, end_date: DATE, goals_type: VARCHAR(100), value_diff: DECIMAL(5,2), user_id: INT, macro_id: INT)

Primary Key: goal_id
Candidate Key: goal_id
ForeignKey: **user_id** references User(user_id), **macro_id** references Macro(macro_id)

Constraints:
user_id NOT NULL.
macro_id NOT NULL and UNIQUE

ExerciseRecord (exercise_record_id: INT, date: DATE, calories_burned: INT, type: VARCHAR(10), user_id: INT)

Primary Key: exercise_record_id
Candidate Key: exercise_record_id
ForeignKey: user_id references User(user_id)
Constraints: user_id NOT NULL

University of British Columbia, Vancouver

Department of Computer Science

Recipe(recipe_id: INT, name: VARCHAR(25))

Primary Key: recipe_id
Candidate Key: recipe_id
ForeignKey:
Constraints: name NOT NULL

Consumes (recipe_id: INT, user_id: INT,)

Primary Key: (recipe_id, user_id)
Candidate Key: (recipe_id, user_id)
ForeignKey: recipe_id references Recipe(recipe_id), user_id references User(user_id)
Constraints:

Step (step_number: INT, description: VARCHAR(55), recipe_id: INT)

Primary Key: (step_number, recipe_id)
Candidate Key: (step_number, recipe_id)
ForeignKey: recipe_id references Recipe(recipe_id)
Constraints:

User (weight: DECIMAL(5,2), height: DECIMAL(5,2), user_id: INT, username: VARCHAR(15), gender: VARCHAR(15), age: INT, recommended_calorie_intake: INT)

Primary Key: user_id
Candidate Keys: user_id, username
ForeignKey:
Constraints: username UNIQUE and NOT NULL

BodyBuilder(training_intensity: DECIMAL(5,2), training_duration_months: INT, body_builder_id: INT)

Primary Key: body_builder_id
Candidate Key: body_builder_id
ForeignKey: body_builder_id references User(user_id)
Constrain:

DiabeticPerson(insulin_level: DECIMAL(5,2), diabetic_person_id: INT, recommended_sugar_intake: INT)

Primary Key: diabetic_person_id
Candidate Key: diabetic_person_id
ForeignKey: diabetic_person_id references User(user_id)
Constraints:

5. Functional Dependencies

User:

- username → user_id, weight, height, gender, age, recommended_calorie_intake
- user_id → username, weight, height, gender, age, recommended_calorie_intake
- (body, weight, height, age) → recommended_calorie_intake

DiabeticPerson:

- insulin_level → recommended_sugar_intake
- diabetic_person_id → insulin_level, recommended_sugar_intake

Macros:

- macro_id → fiber, sugar, fat, carbohydrates, protein, calories

University of British Columbia, Vancouver

Department of Computer Science

Ingredient:

- (name, brand) → taste, food_type, macro_id
- name → food type
 - Rationale: name doesn't determine taste because we want taste to be brand specific (e.g. one brand saltier than the other)

MealRecord:

- meal_record_id → date, user_id, recipe_id

Goals:

- goal_id → start_date, end_date, goals_type, value_diff, user_id, macro_id

ExerciseRecord:

- exercise_record_id → date, calories_burned, type, user_id

Recipe:

- recipe_id → name

Step:

- (step_number, recipe_id) → description

BodyBuilder:

- body_builder_id → training_intensity, training_duration_months
- training_duration_months → training_intensity

Consumes:

- Only trivial functional dependencies

Fails 3nf:

- **DiabeticPerson:**
 - insulin_level → recommended_sugar_intake
- **User:**
 - height, weight, age, gender → recommended_macro_intake
- **BodyBuilder**
 - training_timeline → training_intensity
- **Ingredient**
 - name → food_type

6. Normalization

- **Note:** Decompose each table separately (e.g. DiabeticPerson assumes User hasn't been decomposed yet)

Decomposition of DiabeticPerson:

Original Table:

DiabeticPerson(insulin_level: DECIMAL(5,2), diabetic_person_id: INT, recommended_sugar_intake: INT)

Primary Key: diabetic_person_id
Candidate Key: diabetic_person_id
ForeignKey: diabetic_person_id references User(user_id)
Constraints:

Functional Dependencies:

- insulin_level → recommended_sugar_intake
- diabetic_person_id → insulin_level, recommended_sugar_intake

Closures:

- {diabetic_person_id} += {diabetic_person_id, insulin_level, recommended_sugar_intake}

University of British Columbia, Vancouver

Department of Computer Science

- {insulin_level} += {recommended_sugar_intake, insulin_level}



Decomposed tables:

RecommendedSugarByInsulin(insulin_level, recommended_sugar_intake)

DiabeticPersonInfo(diabetic_person_id, insulin_level)

DiabeticPersonInfo(diabetic_person_id: INT, insulin_level: DECIMAL(5,2))

Primary Key: diabetic_person_id

Candidate Key: diabetic_person_id

ForeignKey: **diabetic_person_id** references User(user_id),
insulin_level references RecommendedSugarByInsulin(insulin_level)

RecommendedSugarByInsulin (insulin_level: DECIMAL(5,2), recommended_sugar_intake: INT)

Primary Key: insulin_level

Candidate Key: insulin_level

Decomposition of User

Original Table:

User (weight: DECIMAL(5,2), height: DECIMAL(5,2), user_id: INT, username: VARCHAR(15), gender: VARCHAR(15), age: INT, recommended_calorie_intake: INT)

Primary Key: user_id

Candidate Keys: user_id, username

ForeignKey:

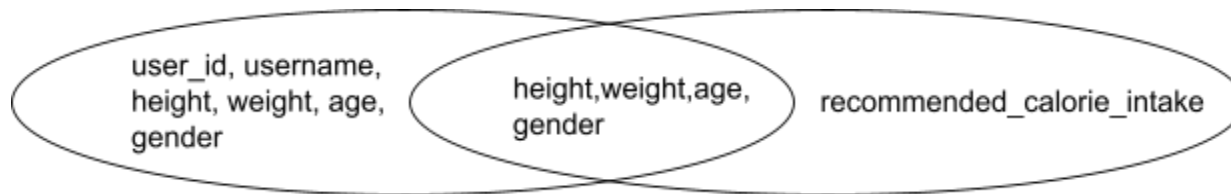
Constraints: username UNIQUE and NOT NULL

Functional Dependencies

- username \rightarrow user_id, weight, height, gender, age, recommended_calorie_intake
- user_id \rightarrow username, weight, height, gender, age, recommended_calorie_intake
- (gender, weight, height, age) \rightarrow recommended_calorie_intake

Closures

- {user_id} += {user_id, username, height, weight, age, gender, recommended_calorie_intake}
- {username} += {user_id, username, height, weight, age, gender, recommended_calorie_intake}
- {height, weight, age, gender} += {height, weight, age, gender, recommended_calorie_intake}



Decomposed Tables:

UserInfo(user_id: INT, username: VARCHAR(15), weight: DECIMAL(5,2), height: DECIMAL(5,2), gender: VARCHAR(15), age: INT)

University of British Columbia, Vancouver

Department of Computer Science

Primary Key: user_id
Candidate Keys: user_id, username
ForeignKey: (height, weight, gender, age) references RecommendedCaloriesForBodyType(height, weight, gender, age)
Constraints: username UNIQUE and NOT NULL

RecommendedCaloriesForBodyType(weight: DECIMAL(5,2), height: DECIMAL(5,2), gender: VARCHAR(15), age: INT, recommended_calorie_intake: INT)

Primary Key: (height, weight, gender, age)
Candidate Key: (height, weight, gender, age)
ForeignKey:
Constraints:

Decomposition of BodyBuilder

Original Table:

BodyBuilder(training_intensity: DECIMAL(5,2), training_duration_months: INT, body_builder_id: INT)

Primary Key: body_builder_id
Candidate Key: body_builder_id
ForeignKey: body_builder_id references User(user_id)
Constrain:

Functional Dependencies

- body_builder_id → training_intensity, training_duration_months
- training_duration_months → training_intensity

Closures

- {body_builder_id} += {body_builder_id, training_intensity, training_duration_months}
- {training_duration_months} += {training_duration_months, training_intensity}



Decomposed Tables:

BodyBuilderInfo (body_builder_id, training_duration_months)

RecommendedTrainingIntensity (training_duration_months, training_intensity)

BodyBuilderInfo (body_builder_id: INT, training_duration_months: INT,)

Primary Key: body_builder_id

Candidate Key: body_builder_id

ForeignKey: **body_builder_id** references User(user_id),

training_duration_months references RecommendedTrainingIntensity (training_duration_months)

RecommendedTrainingIntensity (training_duration_months: INT, training_intensity: DECIMAL(5,2))

Primary Key: training_duration_months

Candidate Key: training_duration_months

Decomposition of Ingredient

Original Table:

Ingredient (name: VARCHAR(255), brand: VARCHAR(50), taste: VARCHAR(255), food_type: VARCHAR(50), macro_id: INT);

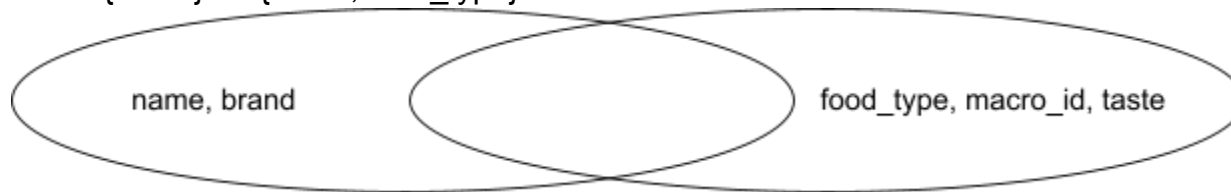
Primary Key: (name, brand)
Candidate Key: (name, brand)
ForeignKey: macro_id references Macros(macro_id)
Constraints: macro_id NOT NULL and UNIQUE

Functional Dependencies

- (name, brand) \rightarrow taste, food_type, macro_id
- name \rightarrow food type

Closures

- {name, brand} += {name, brand, taste, food_type, macro_id}
- {name} += {name, food_type}



Decomposed Tables:

IngredientInfo (name: VARCHAR(255), food_type: VARCHAR(50))

Primary Key: name
Candidate Key: name
ForeignKey:
Constraints:

University of British Columbia, Vancouver

Department of Computer Science

IngredientByBrand (name: VARCHAR(255), brand: VARCHAR(50), taste: VARCHAR(255), macro_id: INT)

Primary Key: (name, brand)

Candidate Key: (name, brand)

ForeignKey: **macro_id** references Macros(macro_id), **name** references IngredientInfo(name)

Constraints: macro_id NOT NULL and UNIQUE

Already in 3nf

Macros (macro_id: INT, fiber: INT, sugar: INT, fat: INT, carbohydrates: INT, protein: INT, calories: INT)

Primary Key: macro_id

Candidate Key: macro_id

ForeignKey:

MealRecord (meal_record_id: INT, date: DATE, user_id: INT, recipe_id: INT)

Primary Key: meal_record_id

Candidate Key: meal_record_id

ForeignKey: **user_id** references User(user_id), **recipe_id** references Recipe(recipe_id)

Constraints: **user_id** NOT NULL, **recipe_id** NOT NULL

University of British Columbia, Vancouver

Department of Computer Science

Goals(goal_id: INT, start_date: DATE, end_date: DATE, goals_type: VARCHAR(100), value_diff: DECIMAL(5,2), user_id: INT, macro_id: INT)

Primary Key: goal_id
Candidate Key: goal_id
ForeignKey: **user_id** references User(user_id), **macro_id** references Macro(macro_id)

Constraints:
user_id NOT NULL.
macro_id NOT NULL and UNIQUE

ExerciseRecord (exercise_record_id: INT, date: DATE, calories_burned: INT, type: VARCHAR(10), user_id: INT)

Primary Key: exercise_record_id
Candidate Key: exercise_record_id
ForeignKey: user_id references User(user_id)
Constraints: user_id NOT NULL

Recipe(recipe_id: INT, name: VARCHAR(25))

Primary Key: recipe_id
Candidate Key: recipe_id
ForeignKey:
Constraints: name NOT NULL

University of British Columbia, Vancouver

Department of Computer Science

Consumes (recipe_id: INT, user_id: INT,)

Primary Key: (recipe_id, user_id)

Candidate Key: (recipe_id, user_id)

ForeignKey: **recipe_id** references Recipe(recipe_id), **user_id** references User(user_id)

Constraints:

Step (step_number: INT, description: VARCHAR(55), recipe_id: INT)

Primary Key: (step_number, recipe_id)

Candidate Key: (step_number, recipe_id)

ForeignKey: recipe_id references Recipe(recipe_id)

Constraints:

7. Final SQL DDL Statements

- Note: Oracle doesn't support ON UPDATE, but we will include it for grading purposes

```
CREATE TABLE IngredientInfo (  
  name VARCHAR(255),  
  food_type VARCHAR(50),  
  PRIMARY KEY(name),  
)
```

```
CREATE TABLE IngredientByBrand (  
  name VARCHAR(255),  
  brand VARCHAR(50),  
  taste VARCHAR(255),  
  macro_id INT NOT NULL,  
  PRIMARY KEY(name, brand),
```

University of British Columbia, Vancouver

Department of Computer Science

```
    FOREIGN KEY(macro_id) REFERENCES Macros (macro_id),  
    FOREIGN KEY(name) REFERENCES IngredientInfo (name),  
    UNIQUE macro_id  
)
```

```
CREATE TABLE Macros (  
    macro_id INT,  
    fiber INT,  
    sugar INT,  
    fat INT,  
    carbohydrates INT,  
    protein INT,  
    calories INT,  
    PRIMARY KEY(macro_id)  
);
```

```
CREATE TABLE MealRecord (  
    meal_record_id INT,  
    date DATE,  
    user_id INT NOT NULL,  
    recipe_id INT NOT NULL,  
    PRIMARY KEY(meal_record_id),  
    FOREIGN KEY (user_id) REFERENCES UserInfo(user_id)  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE,  
    FOREIGN KEY (recipe_id) REFERENCES Recipe(recipe_id)  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE  
)
```

```
CREATE TABLE Goals(  
    goal_id INT,
```

University of British Columbia, Vancouver

Department of Computer Science

```
start_date DATE,  
end_date DATE,  
goals_type VARCHAR(100),  
value_diff DECIMAL(5,2),  
PRIMARY KEY(goal_id),  
user_id INT NOT NULL,  
macro_id INT NOT NULL,  
UNIQUE macro_id,  
FOREIGN KEY(user_id) REFERENCES UserInfo (user_id)  
    ON DELETE SET NULL  
    ON UPDATE CASCADE,  
FOREIGN KEY(macro_id) REFERENCES Macro(macro_id)  
)
```

```
CREATE TABLE ExerciseRecord (  
    exercise_record_id INT,  
    date DATE,  
    calories_burned INT,  
    type VARCHAR(10),  
    user_id INT NOT NULL,  
    PRIMARY KEY(exercise_record_id),  
    FOREIGN KEY (user_id)  
        REFERENCES UserInfo (user_id)  
        ON DELETE NO ACTION  
        ON UPDATE CASCADE  
)
```

```
CREATE TABLE Recipe(  
    recipe_id INT,  
    name VARCHAR(25) NOT NULL,  
    PRIMARY KEY(recipe_id)  
)
```

University of British Columbia, Vancouver

Department of Computer Science

```
CREATE TABLE Consumes (  
    recipe_id INT,  
    user_id INT,  
    PRIMARY KEY(recipe_id, user_id),  
    FOREIGN KEY(recipe_id) REFERENCES Recipe(recipe_id),  
    FOREIGN KEY(user_id) REFERENCES UserInfo (user_id)  
)
```

```
CREATE TABLE Step (  
    step_number INT,  
    description VARCHAR(55),  
    recipe_id INT,  
    PRIMARY KEY(step_number, recipe_id),  
    FOREIGN KEY(recipe_id)  
        REFERENCES Recipe(recipe_id)  
        ON DELETE CASCADE  
)
```

```
CREATE TABLE UserInfo (  
    user_id INT,  
    username VARCHAR(15) UNIQUE NOT NULL,  
    weight DECIMAL(5,2),  
    height DECIMAL(5,2),  
    gender VARCHAR(15),  
    age INT,  
    PRIMARY KEY (user_id)  
    FOREIGN KEY (height, weight, gender, age)  
        REFERENCES RecommendedCaloriesForBodyType(height, weight, gender, age)  
)
```

```
CREATE TABLE RecommendedCaloriesForBodyType (  
    weight DECIMAL(5,2),
```

University of British Columbia, Vancouver

Department of Computer Science

```
height DECIMAL(5,2),  
gender VARCHAR(15),  
age INT,  
recommended_calorie_intake INT,  
PRIMARY KEY (height, weight, gender, age)  
)
```

```
CREATE TABLE BodyBuilderInfo (  
    body_builder_id INT,  
    training_duration_months INT,  
    PRIMARY KEY (body_builder_id ),  
    FOREIGN KEY (body_builder_id) REFERENCES User(user_id),  
    FOREIGN KEY (training_duration_months )  
        REFERENCES RecommendedTrainingIntensity (training_duration_months)  
)
```

```
CREATE TABLE RecommendedTrainingIntensity (  
    training_duration_months INT,  
    training_intensity DECIMAL(5,2),  
    PRIMARY KEY (training_duration_months)  
)
```

```
CREATE TABLE DiabeticPersonInfo (  
    diabetic_person_id INT,  
    insulin_level DECIMAL(5,2),  
    PRIMARY KEY (diabetic_person_id),  
    FOREIGN KEY (diabetic_person_id) REFERENCES UserInfo (user_id),  
    FOREIGN KEY (insulin_level) REFERENCES RecommendedSugarByInsulin(insulin_level)  
)
```

```
CREATE TABLE RecommendedSugarByInsulin (  
    insulin_level DECIMAL(5,2),  
    recommended_sugar_intake INT,
```

University of British Columbia, Vancouver

Department of Computer Science

PRIMARY KEY (insulin_level)
)

8. Populating the tables

IngredientInfo

```
INSERT INTO IngredientInfo (name, food_type)
VALUES ('Tomato', 'Vegetable'),
       ('Chicken Breast', 'Meat'),
       ('Parmesan Cheese', 'Dairy'),
       ('Basil', 'Herb'),
       ('Olive Oil', 'Fat');
```

IngredientByBrand

```
INSERT INTO IngredientByBrand (name, brand, taste, macro_id)
VALUES ('Tomato', 'Campbell's', 'sweeter than average', 1),
       ('Chicken Breast', 'Western Family', 'drier than other brands', 2),
       ('Parmesan Cheese', 'Western Family', 'more fragrant than average', 3),
       ('Basil', 'Uncle Ben's', 'looks fresher than average', 4),
       ('Olive Oil', 'Las Espadas', 'more viscous compared to others', 5);
```

BodyBuilderInfo

```
INSERT INTO BodyBuilderInfo (body_builder_id, training_duration_months)
VALUES (1, 7),
       (2, 8),
       (3, 9),
       (4, 10),
       (5, 11);
```

RecommendedTrainingIntensity

```
INSERT INTO RecommendedTrainingIntensity (training_duration_months, training_intensity)
```

University of British Columbia, Vancouver

Department of Computer Science

```
VALUES (7, 50),  
        (8, 60),  
        (9, 70),  
        (10, 80),  
        (11, 90);
```

Macros

```
INSERT INTO Macros (macro_id, fiber, sugar, fat, carbohydrates, protein, calories)  
VALUES (1, 5, 10, 8, 25, 12, 250),  
        (2, 3, 5, 10, 30, 15, 300),  
        (3, 4, 12, 7, 22, 11, 275),  
        (4, 6, 9, 9, 28, 13, 290),  
        (5, 2, 4, 5, 18, 8, 200);
```

MealRecord

```
INSERT INTO MealRecord (meal_record_id, date, user_id, recipe_id)  
VALUES (1, '2024-10-15', 1, 1),  
        (2, '2024-10-12', 2, 2),  
        (3, '2024-10-11', 3, 4),  
        (4, '2024-10-10', 1, 5),  
        (5, '2024-10-09', 2, 1);
```

Goals

```
INSERT INTO Goals (goal_id, start_date, end_date, goals_type, value_diff, user_id, macro_id)  
VALUES (1, '2024-10-01', '2024-12-31', 'Weight Loss', 5.50, 1, 1),  
        (2, '2024-10-01', '2024-12-31', 'Weight Loss', 5.50, 2, 2),  
        (3, '2024-10-01', '2024-12-31', 'Weight Loss', 5.60, 3, 3),  
        (4, '2024-10-01', '2024-12-31', 'Weight Gain', 5.70, 4, 4),  
        (5, '2024-10-01', '2024-12-31', 'Weight Gain', 5.50, 5, 5);
```

ExerciseRecord

```
INSERT INTO ExerciseRecord (exercise_record_id, date, calories_burned, type, user_id)  
VALUES (1, '2024-10-15', 300, 'Running', 1),  
        (2, '2024-10-11', 300, 'Walking', 2),
```

University of British Columbia, Vancouver

Department of Computer Science

```
(3, '2024-10-12', 500, 'Weight Lifting', 3),  
(4, '2024-10-13', 700, 'Running', 4),  
(5, '2024-10-14', 100, 'Swimming', 5),
```

Recipe

```
INSERT INTO Recipe (recipe_id, name)  
VALUES  
  (1, 'Chicken Salad'),  
  (2, 'Protein Smoothie'),  
  (3, 'Fried Rice'),  
  (4, 'Steak Frites'),  
  (5, 'Fettucine Alfredo')
```

Consumes

```
INSERT INTO Consumes (recipe_id, user_id)  
VALUES  
  (1, 1),  
  (2, 2),  
  (3, 3),  
  (4, 4),  
  (5, 5);
```

Step

```
INSERT INTO Step (step_number, description, recipe_id)  
VALUES  
  (1, 'Marinate chicken breasts', 1),  
  (2, 'Grill the chicken until cooked.', 1),  
  (3, 'Chop vegetables for the salad.', 1),  
  (4, 'Mix all the ingredient.', 1);
```

```
INSERT INTO Step (step_number, description, recipe_id)  
VALUES  
  (1, 'Add frozen fruits to the blender', 2),
```


University of British Columbia, Vancouver

Department of Computer Science

```
(2,'Add water, protein powder,milk', 2),  
(3,'Blend until smooth', 2);
```

```
INSERT INTO Step (step_number, description,recipe_id)  
VALUES
```

```
(1 , 'Cook rice, or use day-old rice', 3),  
(2,'stir-fry diced vegetables and protein in a hot pan.', 3),  
(3,'then season with soy sauce', 3);
```

```
INSERT INTO Step (step_number, description,recipe_id)  
VALUES
```

```
(1 , 'Season steak, cook to preference.', 4),  
(2,'Fry potatoes until crispy.', 4),  
(3,'Butter bath the steak', 4);
```

```
INSERT INTO Step (step_number, description,recipe_id)  
VALUES
```

```
(1 , 'Cook pasta until al dente.', 5),  
(2,'Simmer cream, add cheese.', 5),  
(3,'Toss pasta in sauce.', 5);
```

UserInfo

```
INSERT INTO UserInfo(user_id, username, weight, height, gender, age)  
VALUES
```

```
(1, John Doe the 1st', 70.00, 175, ' Male', 30),  
(2, John Doe the 2nd', 80.00, 180, ' Male', 27),  
(3, John Doe the 3rd', 90.00, 177, ' Male', 22),  
(4, John Doe the 4th', 77.00, 185, ' Male', 26),  
(5, Janine Doe the 5th', 60.00, 170, ' Female, 32);
```

RecommendedCaloriesForBodyType

University of British Columbia, Vancouver

Department of Computer Science

```
INSERT INTO RecommendedCaloriesForBodyType (weight, height, gender, age, recommended_calorie_intake)
VALUES (70.00, 175, ' Male', 30, 2500),
      (80.00, 180, ' Male', 27, 2600),
      (90.00, 177, ' Male', 22, 2700),
      (77.00, 185, ' Male', 26, 2600),
      (60.00, 170, ' Female', 32, 2000);
```

DiabeticPersonInfo

```
INSERT INTO DiabeticPersonInfo (diabetic_person_id, insulin_level)
VALUES
  (1, 5.00),
  (2, 6.50),
  (3, 7.80),
  (4, 9.20),
  (5, 9.80);
```

RecommendedSugarByInsulin

```
INSERT INTO RecommendedSugarByInsulin (insulin_level, recommended_sugar_intake)
VALUES
  (5.00, 55),
  (6.50, 45),
  (7.80, 35),
  (9.20, 25),
  (9.80, 20);
```