University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: Feb. 29, 2024

Group Number: 40

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Ubada Raja	99035578	m4p5v	ubada@student.ubc.ca
Nam Nguyen	89939383	e5n5h	nnguye10@student.ubc.ca
Hieu Le	76067321	s3b9t	xle@student.ubc.ca

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.)

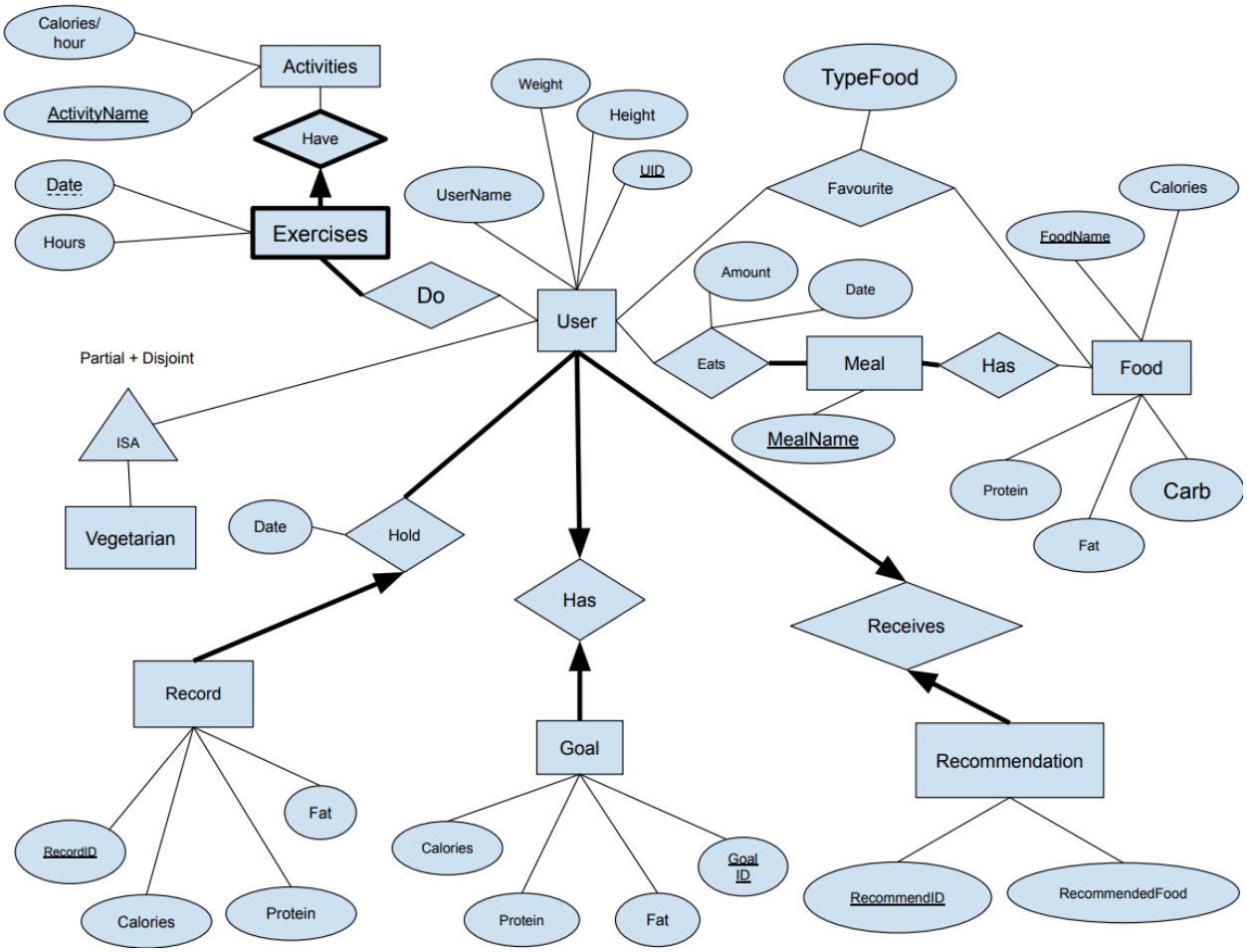
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

2. Brief summary

- Our project helps individuals maintain a balanced diet by tracking their daily food intake, and exercise/ activities level. It also provides recommendations to help them reach their goals.

3. ER Diagram: (same as Milestone 1)

- We removed the total participation from “Has” to “Food”, “Food” can appear in our database (Pasta, beef, broccoli, etc), but doesn’t necessarily need to appear in any meal in the “Meal” table
- We added the attributes “Amount” and “Date” to Meal
- We added the attributes “TypeFood” to Favourite
- We just renamed the attribute to make it easier to distinguish later.
    + Name in “User” to UserName
    + Name in “Activities” to ActivityName
    + Name in “Meal” to MealName
    + Name in “Food” to FoodName



4. The schema derived from ER diagram

a. Table definition (Table1(attr1: domain1 …etc.)). Include domains for each attr
b. Specify the primary key (PK underlined) candidate key (CK), foreign key (FK bolded) and other constraints (not null, unique),

- User(UID: char[10], UserName: varchar, Weight: integer, Height: integer, **GoalID**: char[10], **RecommendID**: char[10]) (GoalID references Goal, RecommendID references Recommendation) (Weight and Height is NOT NULL)
- Activities (CaloriesPerHour: integer, ActivityName: varchar) (CaloriesPerHour is NOT NULL)
- Exercise_Activities(Date: date, **ActivityName**: varchar, Hour: integer) (ActivityName references Activities) (Hour is NOT NULL)
- Food (FoodName: varchar, Protein: integer, Calorie: integer, Fat: integer) (Protein, Fat, Calorie NOT NULL)
- Record_Hold (RecordID: char[10], **UID**: char[10], Date: date, Calories: integer, Protein: integer, Fat: integer) (UID references User) (Protein, Fat, Calorie, Date, UID NOT NULL)
- Goal (GoalID: char[10], Calories: integer, Protein: integer, Fat: integer) (Protein, Fat, Calorie NOT NULL)
- Recommendation (RecommendID: char[10], **RecommendedFood**: varchar) (RecommendFood references Food) (RecommendedFood NOT NULL)
- Vegetarian (**UID:** char[10]) (UID references User)
- Eats_Meal (**UID**: char[10], **MealName**: varchar, Date: date, Amount: integer) (UID references User, MealName references Meal_Has_Food) (MealName is NOT NULL)
- Meal_Has_Food (MealName: varchar, **FoodName**: varchar) (FoodName references Food)
- Favourite (**UID**: char[10], **FoodName**: varchar, TypeFood: varchar) (UID references User, FoodName references Food)
- User_Do_Exercise (**UID:** char[10], **Date**: date, **ActivityName**: varchar) (UID references User, Date references Exercises, ActivityName references Activities) (Date and Activity Name are NOT NULL)

5. Functional Dependencies

User:

UID -> UserName, Weight, Height, GoalID, RecommendID

Activities:

ActivityName -> CaloriesPerHour

Exercise_Activities:

Date, ActivityName -> Hour

Food:

FoodName -> Calories, Protein, Fat, Carb

Protein, Fat, Carb -> Calories

Record_Hold:

RecordID -> Calories, Protein, Fat

Goal:

GoalID -> Calories, Protein, Fat

Recommendation:

RecommendID -> RecommendedFood

Vegetarian:

Eats_Meal:

UID, MealName, Date -> Amount

Meal_Has_Food:

Favorite:

UID, FoodName -> TypeFood

FoodName -> TypeFood

User_Do_Exercise:

6. Normalization

- Normalization Favorite(UID, FoodName, TypeFood)

Left	Middle	Right
UID, FoodName	TypeFood	

{UID} += {UID}
{FoodName} += {FoodName, TypeFood}
{UID, FoodName} += {UID, FoodName, TypeFood}

FD: FoodName -> TypeFood violates BCNF
R1(UID, FoodName) R2(FoodName, TypeFood)
Final Answer: R1(UID, FoodName), R2(FoodName, TypeFood)

- Normalization Food:
R(FoodName, Calories,Protein,Fat,Carb)
FDs: FoodName -> Calories, Protein, Fat, Carb
Protein, Fat, Carb -> Calories

Left	Middle	Right
FoodName	Calories, Protein, Fat, Carb	

{FoodName}+ = {FoodName, Calories, Protein, Fat, Carb}
{Protein, Fat, Carb}+ = {Protein, Fat, Carb, Calories}
FD: Protein, Fat, Carb -> Calories violates BCNF
=> decompose into R1(Protein,Fat,Carb,Calories)
and R2(Protein, Fat, Carb, FoodName)
Final answer:
R1(Protein,Fat,Carb,Calories) and R2(Protein, Fat, Carb, FoodName)

- User(UID: char[10], UserName: varchar, Weight: integer, Height: integer, **GoalID**: char[10], **RecommendID**: char[10]) (GoalID references Goal, RecommendID references Recommendation) (Weight and Height is NOT NULL)
- Activities (CaloriesPerHour: integer, ActivityName: varchar) (CaloriesPerHour is NOT NULL)
- Exercise_Activities(Date: date, **ActivityName**: varchar, Hour: integer) (ActivityName references Activities) (Hour is NOT NULL)

- Nutrition(Protein: integer, Fat: integer, Carb: integer, Calories: integer) (Protein, Fat, Calorie NOT NULL)
- FoodNutrition(Protein: integer, Fat: integer, Carb: integer, FoodName: varchar) (Protein, Fat, Calorie NOT NULL)
- Record_Hold (RecordID: char[10], **UID**: char[10], Date: date, Calories: integer, Protein: integer, Fat: integer) (UID references User) (Protein, Fat, Calorie, Date, UID NOT NULL)
- Goal (GoalID: char[10], Calories: integer, Protein: integer, Fat: integer) (Protein, Fat, Calorie NOT NULL)
- Recommendation (RecommendID: char[10], **RecommendedFood**: varchar) (RecommendFood references Food) (RecommendedFood NOT NULL)
- Vegetarian (**UID:** char[10]) (UID references User)
- Eats_Meal (**UID**: char[10], **MealName**: varchar, Date: date, Amount: integer) (UID references User, MealName references Meal_Has_Food) (MealName is NOT NULL)
- Meal_Has_Food (MealName: varchar, **FoodName**: varchar) (FoodName references Food)
- Favourite_1 (**UID**: char[10], FoodName: varchar) (UID references User)
- Favourite_2 (**FoodName**: varchar, TypeFood: varchar) (FoodName references Food)
- User_Do_Exercise (**UID:** char[10], **Date**: date, **ActivityName**: varchar) (UID references User, Date references Exercises, ActivityName references Activities) (Date and Activity Name are NOT NULL)

7. SQL DDL statements

```
CREATE TABLE User(
        UID CHAR(10),
        Name VARCHAR,
        Weight INTEGER NOT NULL,
        Height INTEGER NOT NULL,
        GoalID CHAR(10) DEFAULT '0000000000',
        RecommendID CHAR(10) DEFAULT '0000000000',
        PRIMARY KEY(UID),
        FOREIGN KEY(GoalID)
                REFERENCES Goal(GoalID)
                ON DELETE SET DEFAULT
                ON UPDATE CASCADE,
        FOREIGN KEY(RID)
                REFERENCES Recommendation(RecommendID)
                ON DELETE SET DEFAULT
                ON UPDATE CASCADE,
        UNIQUE GoalID,
        UNIQUE RecommendID);

CREATE TABLE Vegetarian(
        UID CHAR(10)
        PRIMARY KEY(UID),
        FOREIGN KEY(UID)
                REFERENCES User(UID)
                ON DELETE CASCADE
        );

CREATE TABLE Activities(
        ActivityName VARCHAR,
        CaloriesPerHour INTEGER NOT NULL,
        PRIMARY KEY(Name));

CREATE TABLE Exercise_Activities(
        Date DATE,
        ActivityName VARCHAR,
        Hours INTEGER NOT NULL,
```

```
        PRIMARY KEY(Date, ActivityName)),
        FOREIGN KEY(ActivityName)
                REFERENCES(Activity(ActivityName))
                ON UPDATE CASCADE
                ON DELETE CASCADE);

CREATE TABLE Nutrition(
        Protein INTEGER NOT NULL,
        Fat INTEGER NOT NULL,
        Carb INTEGER NOT NULL,
        Calorie INTEGER NOT NULL,
        PRIMARY KEY(Protein,Fat,Carb));

CREATE TABLE FoodNutrition(
        Protein INTEGER NOT NULL,
        Fat INTEGER NOT NULL,
        Carb INTEGER NOT NULL,
        FoodName VARCHAR,
        PRIMARY KEY(FoodName));
);

CREATE TABLE Record_Hold(
        RecordID CHAR(10),
        UID CHAR(10) NOT NULL,
        Date DATE NOT NULL,
        Calories INTEGER NOT NULL,
        Protein INTEGER NOT NULL,
        Fat INTEGER NOT NULL,
        PRIMARY KEY(RecordID),
        FOREIGN KEY(UID)
                REFERENCES User(UID)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

CREATE TABLE Goal(
        GoalID CHAR(10),
        Calories INTEGER NOT NULL,
```

```
        Protein INTEGER NOT NULL,
        Fat INTEGER NOT NULL,
        PRIMARY KEY(GoalID),
);

CREATE TABLE Recommendation(
        RecommendID CHAR(10),
        RecommendedFood VARCHAR NOT NULL,
        PRIMARY KEY(RecommendID),
);
CREATE TABLE Eats_Meal(
        UID CHAR(10),
        MealName VARCHAR,
        Date DATE,
        Amount INTEGER,
        PRIMARY KEY(UID, MealName,Date),
        FOREIGN KEY(UID)
                REFERENCES User(UID)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY(MealName)
                REFERENCES Meal_Has_Food(MealName)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

CREATE TABLE Meal_Has_Food(
        MealName VARCHAR,
        FoodName VARCHAR,
        PRIMARY KEY(MealName, FoodName),
        FOREIGN KEY(FoodName)
                REFERENCES Meal(FoodName)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

CREATE TABLE Favourite_1(
        UID CHAR(10),
```

```
        FoodName VARCHAR,
        PRIMARY KEY(UID),
        FOREIGN KEY(UID)
                REFERENCES User(UID)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

CREATE TABLE Favourite_2 (
        FoodName VARCHAR,
        TypeFood VARCHAR,
        PRIMARY KEY(FoodName),
        FOREIGN KEY(FoodName)
                REFERENCES Food(FoodName)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

CREATE TABLE User_Do_Exercise(
        UID CHAR(10),
        Date DATE,
        ActivityName VARCHAR,
        PRIMARY KEY(Date,  UID, ActivityName),
        FOREIGN KEY(Date)
                REFERENCES Exercise(Date)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY(UID)
                REFERENCES User(UID)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY(ActivityName)
                REFERENCES Activities(ActivityName)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);
```

8. INSERT statements

INSERT INTO User(UID, Name, Weight, Height) VALUES (‘u1’, ‘John’, 70, 167);
INSERT INTO User(UID, Name, Weight, Height) VALUES (‘u2’, ‘Ubada’, 80, 170);
INSERT INTO User(UID, Name, Weight, Height) VALUES (‘u3’, ‘Hieu’, 65, 185);
INSERT INTO User(UID, Name, Weight, Height) VALUES (‘u4’, ‘Nam’, 75, 182);
INSERT INTO User(UID, Name, Weight, Height) VALUES (‘u5’, ‘Mark’, 65, 160);
INSERT INTO User(UID, Name, Weight, Height) VALUES (‘u6’, ‘Jake’, 100, 200);
INSERT INTO User(UID, Name, Weight, Height) VALUES (‘u7’, ‘Raymond’, 50, 150);

INSERT INTO Vegetarian(UID) VALUES(‘u1’);
INSERT INTO Vegetarian(UID) VALUES(‘u6’);
INSERT INTO Vegetarian(UID) VALUES(‘u7’);
INSERT INTO Vegetarian(UID) VALUES(‘u5’);
INSERT INTO Vegetarian(UID) VALUES(‘u3’);

INSERT INTO Activity(Name, CaloriesPerHour) Values(‘Cycling’, 300);
INSERT INTO Activity(Name, CaloriesPerHour) Values(‘Running’, 700);
INSERT INTO Activity(Name, CaloriesPerHour) Values(‘Weightlifting’, 180);
INSERT INTO Activity(Name, CaloriesPerHour) Values(‘Basketball’, 650);
INSERT INTO Activity(Name, CaloriesPerHour) Values(‘Swimming’, 650);

INSERT INTO Exercise_Activities (Date, ActivityName, Hours) VALUES (2024-02-27, ‘Cycling’, 1);
INSERT INTO Exercise_Activities (Date, ActivityName, Hours) VALUES (2024-02-27, ‘Running’, 0.25);
INSERT INTO Exercise_Activities (Date, ActivityName, Hours) VALUES (2024-02-28, ‘Weightlifting, 2);
INSERT INTO Exercise_Activities (Date, ActivityName, Hours) VALUES (2024-03-01, ‘Swimming’, 0.5);
INSERT INTO Exercise_Activities (Date, ActivityName, Hours) VALUES (2024-03-02, ‘Basketball’, 3);

INSERT INTO User_Do_Exercise (UID, Date, ActivityName) VALUES (‘u2’, 2024-02-27, ‘Cycling’);
INSERT INTO User_Do_Exercise (UID, Date, ActivityName) VALUES (‘u3’, 2024-02-27, ‘Running’);
INSERT INTO User_Do_Exercise (UID, Date, ActivityName) VALUES (‘u2’, 2024-02-28, ‘Weightlifting’);
INSERT INTO User_Do_Exercise (UID, Date, ActivityName) VALUES (‘u6’, 2024-03-02, ‘Basketball’);

INSERT INTO User_Do_Exercise (UID, Date, ActivityName) VALUES ('u5', 2024-03-01, 'Swimming');

INSERT INTO Record_Hold(RecordID, UID, Date, Calories, Protein, Fat) VALUES ('r1', 'u1', 2024-03-01, 1500, 90, 25);
INSERT INTO Record_Hold(RecordID, UID, Date, Calories, Protein, Fat) VALUES ('r2', 'u3', 2024-03-01, 2250, 120, 15);
INSERT INTO Record_Hold(RecordID, UID, Date, Calories, Protein, Fat) VALUES ('r3', 'u2', 2024-03-01, 2000, 75, 10);
INSERT INTO Record_Hold(RecordID, UID, Date, Calories, Protein, Fat) VALUES ('r4', 'u5', 2024-03-01, 1850, 50, 30);
INSERT INTO Record_Hold(RecordID, UID, Date, Calories, Protein, Fat) VALUES ('r5', 'u6', 2024-03-01, 2600, 120, 25);

INSERT INTO Goal(GoalID, Calories, Protein, Fat) VALUES ('g1', 2750, 120, 25);
INSERT INTO Goal(GoalID, Calories, Protein, Fat) VALUES ('g2', 2000, 90, 25);
INSERT INTO Goal(GoalID, Calories, Protein, Fat) VALUES ('g3', 1800, 75, 30);
INSERT INTO Goal(GoalID, Calories, Protein, Fat) VALUES ('g4', 2500, 100, 35);
INSERT INTO Goal(GoalID, Calories, Protein, Fat) VALUES ('g5', 3000, 200, 25);

INSERT INTO Recommendation(RecommendID, RecommendedFood) VALUES ('reco1', 'Chicken');
INSERT INTO Recommendation(RecommendID, RecommendedFood) VALUES ('reco2', 'Broccoli');
INSERT INTO Recommendation(RecommendID, RecommendedFood) VALUES ('reco3', 'Beef');
INSERT INTO Recommendation(RecommendID, RecommendedFood) VALUES ('reco4', 'Apple');
INSERT INTO Recommendation(RecommendID, RecommendedFood) VALUES ('reco5', 'Rice');

UPDATE Users
SET GoalID = 'g1', RecommendID = 'reco1'
WHERE UID = 'u1';

UPDATE Users
SET GoalID = 'g2', RecommendID = 'reco2'
WHERE UID = 'u7';

UPDATE Users
SET GoalID = 'g3', RecommendID = 'reco3'
WHERE UID = 'u3';

```
UPDATE Users
SET GoalID = 'g4', RecommendID = 'reco4'
WHERE UID = 'u6';

UPDATE Users
SET GoalID = 'g5', RecommendID = 'reco5'
 WHERE UID = 'u5';

INSERT INTO Favourite_1(UID, FoodName) VALUES  ('u1', 'Broccoli');
INSERT INTO Favourite_1(UID, FoodName) VALUES  ('u2', 'Chicken');
INSERT INTO Favourite_1(UID, FoodName) VALUES  ('u3', 'Apple');
INSERT INTO Favourite_1(UID, FoodName) VALUES  ('u6', 'Tofu');
INSERT INTO Favourite_1(UID, FoodName) VALUES  ('u5', 'Rice');

INSERT INTO Favourite_2(FoodName, TypeFood) VALUES  ('Chicken', 'Meat');
INSERT INTO Favourite_2(FoodName, TypeFood) VALUES  ('Broccoli', 'Vegetable');
INSERT INTO Favourite_2(FoodName, TypeFood) VALUES  ('Apple', 'Fruit');
INSERT INTO Favourite_2(FoodName, TypeFood) VALUES  ('Tofu', 'Vegetable');
INSERT INTO Favourite_2(FoodName, TypeFood) VALUES  ('Rice', 'Vegetable');

INSERT INTO Meal_Has_Food(MealName, FoodName) VALUES  ('Chicken Stir-Fry', 'Chicken');
INSERT INTO Meal_Has_Food(MealName, FoodName) VALUES  ('Chicken Stir-Fry', 'Broccoli');
INSERT INTO Meal_Has_Food(MealName, FoodName) VALUES  ('Tofu & Rice', 'Tofu');
INSERT INTO Meal_Has_Food(MealName, FoodName) VALUES  ('Tofu & Rice', 'Rice');
INSERT INTO Meal_Has_Food(MealName, FoodName) VALUES  ('Grilled Chicken', 'Chicken');
INSERT INTO Meal_Has_Food(MealName, FoodName) VALUES ('Fruit Salad', 'Apple');
INSERT INTO Meal_Has_Food(MealName, FoodName) VALUES ('Fruit Salad', 'Banana');

INSERT INTO Eats_Meal(UID, MealName, Date, Amount) VALUES ('u1' 'Chicken Stir-Fry', 2024-03-01, 250);
INSERT INTO Eats_Meal(UID, MealName, Date, Amount) VALUES ('u3' 'Tofu & Rice', 2024-04-28, 500);
INSERT INTO Eats_Meal(UID, MealName, Date, Amount) VALUES ('u3' 'Fruit Salad', 2024-03-02, 400);
INSERT INTO Eats_Meal(UID, MealName, Date, Amount) VALUES ('u2' 'Grilled Chicken', 2024-03-01, 750);
```

INSERT INTO Eats_Meal(UID, MealName, Date, Amount) VALUES (‘u1’ ‘Fruit Salad’, 2024-03-01, 650);

INSERT INTO Nutrition(Protein, Fat, Carb, Calorie) VALUES (25, 10, 40, 500);
INSERT INTO Nutrition(Protein, Fat, Carb, Calorie) VALUES (85, 70, 20, 1050);
INSERT INTO Nutrition(Protein, Fat, Carb, Calorie) VALUES (2, 70, 10, 678);
INSERT INTO Nutrition(Protein, Fat, Carb, Calorie) VALUES (100, 95, 20, 1335);
INSERT INTO Nutrition(Protein, Fat, Carb, Calorie) VALUES (10, 150, 20, 1050);
INSERT INTO Nutrition(Protein, Fat, Carb, Calorie) VALUES (40, 0, 10, 200);

INSERT INTO FoodNutrition(Protein, Fat, Carb, FoodName) VALUES (0.31, 0.036, 0, ‘Chicken’);
INSERT INTO FoodNutrition(Protein, Fat, Carb, FoodName) VALUES (0.17, 0.09, 0.03, ‘Tofu’);
INSERT INTO FoodNutrition(Protein, Fat, Carb, FoodName) VALUES (0.043, 0.006, 0.032, ‘Broccoli’);
INSERT INTO FoodNutrition(Protein, Fat, Carb, FoodName) VALUES (0.02, 0, 0.28, ‘Rice’);
INSERT INTO FoodNutrition(Protein, Fat, Carb, FoodName) VALUES (0.02, 0, 0.28, ‘Rice’);
INSERT INTO FoodNutrition(Protein, Fat, Carb, FoodName) VALUES (0.003, 0.002, 0.14, ‘Apple’);
INSERT INTO FoodNutrition(Protein, Fat, Carb, FoodName) VALUES (0.011, 0.003, 0.23, ‘Banana’);