

**TIETOKANNAT TTC2020-3029**  
**PRACTICE WORK**  
**RAJESHWARI REDDY DASARIPALLY AD2008**

## **NUTRITIONDATABASE**

### **General Description:**

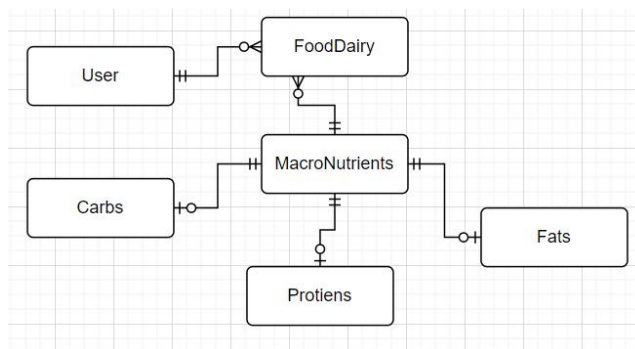
I decided to do Database practice work by single. I selected an imaginary nutrition track of user and the related relational database as my subject. The idea of the database is to have track the user foodlog with macronutrients and calories in food items , with their daily major workouts they do in their life. The mentioned macronutrients and calories for each food item is collected from nutrition websites which also helps me to know nutrient values of the food we intake in our dailylife.

### **TRACKING individual person Nutrition in their dailylife.**

1. The goal is a preliminary plan to track consumption habits from user and also track their major activities they do everyday.
2. To gather personal information from user.
3. User daily consumption into Food log/Food dairy
4. To contain a list of food items and its appropriate calories, macronutrients and details.
5. This data base includes User, Macronutrients, foodlog, Exercise and caloriesgoal.
6. I created a preliminarary ER diagrams using drawio. Which is shown below.

### **Conceptual model 1.0:**

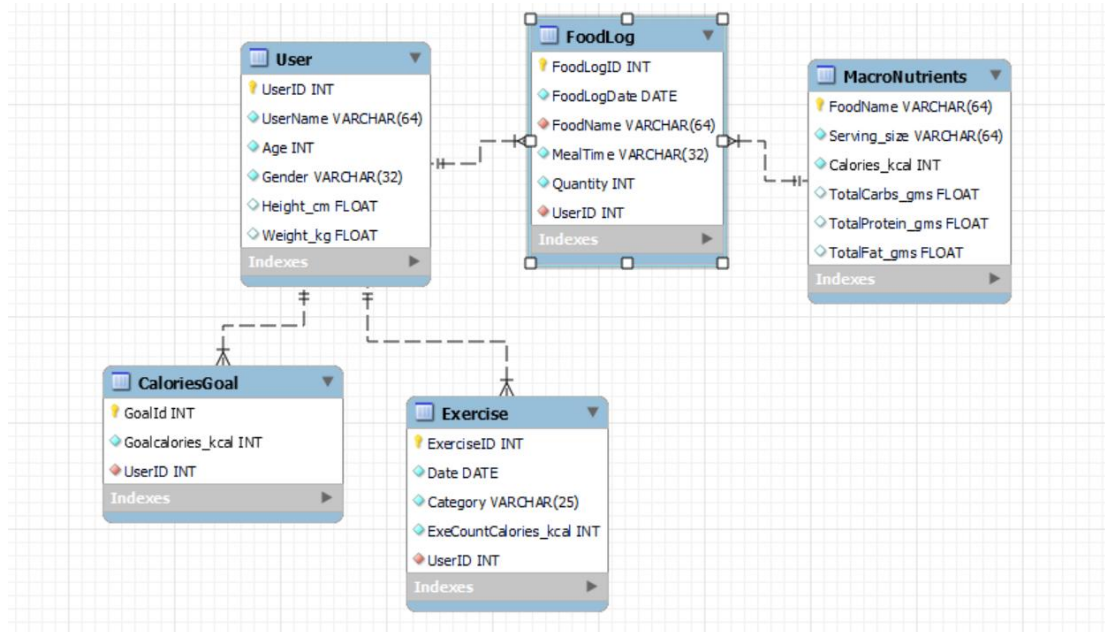
My first conceptual model was shown below , but later I thought to change few entities mentioned in this model.



The final conceptual model is shown as EER diagram in the workbench .

## Conceptual model 2.0:

### EER CHART



### Explanation:

#### Entities and Attributes:

- 1) The User table stores information about the userID, userName, age, gender, height, weight.
- 2) The Foodlog stores information about foodlogID, foodlogDate, foodName, mealTime, quantity.
- 3) The Macronutrient table stores information about different macronutrients (carbohydrates, protein, and fat), including their foodnames, foodportion and calorie counts.
- 4) The Exercise table stores information about different types of exercises, including their names and exercisecountcalorie.
- 5) The CalorieGoal table stores information about each user's calorie goals, including the user ID.

#### Entity Relationships

- 1) A user from the user table may record one or many food items in the foodlog. A food item recorded in the foodlog must be a user in the usertable.
- 2) A foodname from the macronutrients table may or maynot be recorded many times in foodlog. A food item recorded in the foodlog must be from a foodname in the macronutrients.
- 3) A user from the user table may record one or many category in the exercise table. An exercise catogory recorded in the exercise table must be a user in the usertable.
- 4) A user from the user table may record one or many goalcalories in the CaloriesGoal table. A goalcalories recorded in the CaloriesGoal table must be a user in the usertable.

## Nutrition DataBase script:

-- MySQL Workbench Forward Engineering

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
```

-----  
-- Schema Nutrition\_db  
-----

-----  
-- Schema Nutrition\_db  
-----

```
CREATE SCHEMA IF NOT EXISTS `Nutrition_db` ;
USE `Nutrition_db` ;
```

-----  
-- Table `Nutrition\_db`.`User`  
-----

```
CREATE TABLE IF NOT EXISTS `Nutrition_db`.`User` (
  `UserID` INT NOT NULL AUTO_INCREMENT,
  `UserName` VARCHAR(64) NOT NULL,
  `Age` INT NOT NULL,
  `Gender` VARCHAR(32) NOT NULL,
  `Height_cm` FLOAT NULL,
  `Weight_kg` FLOAT NULL,
  PRIMARY KEY (`UserID`))
ENGINE = InnoDB;
```

-----  
-- Table `Nutrition\_db`.`MacroNutrients`  
-----

```
CREATE TABLE IF NOT EXISTS `Nutrition_db`.`MacroNutrients` (
  `FoodName` VARCHAR(64) NOT NULL,
  `Serving_size` VARCHAR(64) NOT NULL,
  `Calories_kcal` INT NOT NULL,
  `TotalCarbs_gms` FLOAT NULL,
  `TotalProtein_gms` FLOAT NULL,
  `TotalFat_gms` FLOAT NULL,
  PRIMARY KEY (`FoodName`))
ENGINE = InnoDB;
```

-----  
-- Table `Nutrition\_db`.`Exercise`  
-----

```
CREATE TABLE IF NOT EXISTS `Nutrition_db`.`Exercise` (
  `ExerciseID` INT NOT NULL AUTO_INCREMENT,
```

```

`Date` DATE NOT NULL,
`Category` VARCHAR(25) NOT NULL,
`ExeCountCalories_kcal` INT NOT NULL,
`UserID` INT NOT NULL,
PRIMARY KEY (`ExerciseID`),
INDEX `fk_Exercise_Customer1_idx` (`UserID` ASC),
CONSTRAINT `fk_Exercise_Customer1`
  FOREIGN KEY (`UserID`)
    REFERENCES `Nutrition_db`.`User` (`UserID`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `UserID`
  FOREIGN KEY (`UserID`)
    REFERENCES `Nutrition_db`.`Exercise` (`UserID`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;

```

```

-----
-- Table `Nutrition_db`.`CaloriesGoal`
-----

```

```

CREATE TABLE IF NOT EXISTS `Nutrition_db`.`CaloriesGoal` (
  `GoalId` INT NOT NULL AUTO_INCREMENT,
  `Goalcalories_kcal` INT NOT NULL,
  `UserID` INT NOT NULL,
  PRIMARY KEY (`GoalId`),
  INDEX `fk_CaloriesGoal_User1_idx` (`UserID` ASC),
  CONSTRAINT `fk_CaloriesGoal_User1`
    FOREIGN KEY (`UserID`)
      REFERENCES `Nutrition_db`.`User` (`UserID`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `UserID`
    FOREIGN KEY (`UserID`)
      REFERENCES `Nutrition_db`.`CaloriesGoal` (`UserID`)
      ON DELETE CASCADE
      ON UPDATE CASCADE)
ENGINE = InnoDB;

```

```

-----
-- Table `Nutrition_db`.`FoodLog`
-----

```

```

CREATE TABLE IF NOT EXISTS `Nutrition_db`.`FoodLog` (
  `FoodLogID` INT NOT NULL AUTO_INCREMENT,
  `FoodLogDate` DATE NOT NULL,
  `FoodName` VARCHAR(64) NOT NULL,
  `MealTime` VARCHAR(32) NOT NULL,
  `Quantity` INT NOT NULL,
  `UserID` INT NOT NULL,
  PRIMARY KEY (`FoodLogID`),
  INDEX `fk_User_has_MacroNutrients_MacroNutrients1_idx` (`FoodName` ASC),
  INDEX `fk_User_has_MacroNutrients_User1_idx` (`UserID` ASC),
  CONSTRAINT `fk_User_has_MacroNutrients_User1`
    FOREIGN KEY (`UserID`)
      REFERENCES `Nutrition_db`.`User` (`UserID`)

```

```

ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_User_has_MacroNutrients_MacroNutrients1`
FOREIGN KEY (`FoodName`)
REFERENCES `Nutrition_db`.`MacroNutrients` (`FoodName`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `FoodName`
FOREIGN KEY (`FoodName`)
REFERENCES `Nutrition_db`.`MacroNutrients` (`FoodName`)
ON DELETE CASCADE
ON UPDATE CASCADE,
CONSTRAINT `UserID`
FOREIGN KEY (`UserID`)
REFERENCES `Nutrition_db`.`User` (`UserID`)
ON DELETE CASCADE
ON UPDATE CASCADE)
ENGINE = InnoDB;

```

```

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

## Adding Test Data:

### 1) Inserting Data in the USER Table

```

Insert into user (UserID, UserName, Age, Gender, Height_cm, Weight_kg)
values (1, 'Lalitha', 25, 'female', 162.56, 45), (2, 'Sureka', 32,
'female', 167.64, 58),
(3, 'Lokesh', 35, 'male', 172.72, 80), (4, 'Karthik', 39, 'male',
182.88, 85);

```

### 2) Inserting Data in the CaloriesGoal Table

```

INSERT INTO CaloriesGoal (GoalCalories_kcal, UserID)
values (2200, 1), (2000, 2), (2600, 3), (2500, 4);

```

### 3) Inserting Data in the Exercise Table

```

ALTER TABLE Exercise auto_increment = 51;

```

```

INSERT INTO Exercise (Date, Category, ExeCountCalories_kcal, UserID)
VALUES ('2023-02-18', 'swimming', 600, 1), ('2023-02-18', 'walking',
350, 2),
('2023-02-18', 'rowing', 800, 3), ('2023-02-18', 'swimming', 500, 4),
('2023-02-19', 'jump roping', 900, 1), ('2023-02-19', 'running', 500,
2),
('2023-02-19', 'walking', 450, 3), ('2023-02-19', 'walking', 400, 4);

```

### 4) Inserting Data in the MacroNutrients Table

```

INSERT INTO MacroNutrients (FoodName, Serving_size, Calories_Kcal,
TotalCarbs_gms, TotalProtein_gms, TotalFat_gms) values
('Whole wheat bread', '1 slice', 56, 11, 2, 0.7),
('Rye Bread', '1 slice', 56, 12, 2, 0.3),
('boiled egg', '1 no', 77, 0.6, 6.3, 5.3),
('Whole grain tortilla', '1 no', 127, 19.0, 4.0, 4.0),

('Milk skim', '1 cup', 86, 11.8, 8, 4),
('Milk lowfat', '1 cup', 121, 11.7, 8, 4.7),
('coffee black without sugar', '1 cup', 3, 0, 0.3, 0.1),
('coffee with milk without sugar', '1 cup', 6, 0.84, 0.33, 0.15),

('Potato-baigan mixed vegetable', '100gms', 100, 32.3, 1.3, 4.7),
('Potato-beans mixed vegetable', '100gms', 134, 34.3, 2.4, 8.7) ,
('Chickpeas', '100gms', 74, 3.3, 4.3, 4.1),
('Vegetable kofta curry', '100gms', 147, 13.3, 2.6, 9.1),
('Dal Makhani', '100gms', 117, 14.0, 6.8, 1.3),

('Boiled rice', '100gms', 111, 25.0, 2.2, 0.2),
('Quinoa cooked', '1 cup', 115, 21.5, 4, 2),

('Salmon', '3 ounces', 121, 0, 17, 5.4),
('Chicken breast', '4 ounces', 193, 0, 29, 7.6),
('grilled fish', '100 gms', 109, 0.28, 22.6, 1.18),

('vegetable soup', '1 cup', 106, 12, 4.12, 4.45),
('Lentil soup', '1 cup', 186, 26.61, 4.12, 4.59 ),

('Yogurt lowfat', '1 cup', 144, 16, 12, 3.5),
('Yogurt nonfat', '1 cup', 127, 10, 11, 5.8),
('Popcorn', '1 cup', 54, 10.7, 2, 0.7),

('chicken burger', '1', 535, 57, 29, 20),
('french fries', '100 gms', 196, 18.5, 1.9, 13.1),
('soft drink', '1 can', 140, 36.09, 0.26, 0.07),
('chicken pizza', '1 piece', 180, 25.47, 7.63, 5 ),

('Orange', '1', 65, 16.3, 1.4, 0.1),
('apple', '1', 81, 21.1, 0.3, 0.5),
('banana', '1', 105, 26.7, 1.2, 0.6),
('blueberries', '1 cup', 82, 20.4, 1.0, 0.6),
('strawberries', '1 cup', 46, 10.4, 1.0, 0.6),
('Pear', '1', 98, 25.1, 0.7, 0.7),
('Watermelon diced', '1 cup', 50, 3.6, 1.0, 0.2),

('icecream', '100 gms', 201, 24.4, 3.52, 10.72),
('dark chocolate', '1 bar', 70, 25, 2, 12),
('biscuits', '45 gms', 166, 19.3, 3.2, 8.5)
;

```

## 5) Inserting Data in the FoodLog Table

```

INSERT INTO foodlog (FoodLogDate, FoodName, MealTime, Quantity,
UserID) values
('2023-02-18', 'apple', 'breakfast', 1, 1), ('2023-02-18', 'Rye
Bread', 'breakfast', 2, 1),
('2023-02-18', 'boiled egg', 'breakfast', 1, 1), ('2023-02-18', 'Milk
skim', 'breakfast', 1, 1),

```

```
('2023-02-18', 'blueberries', 'breakfast', 1, 2), ('2023-02-18',
'Whole wheat bread', 'breakfast', 1, 2),
('2023-02-18', 'Milk lowfat', 'breakfast', 1, 2), ('2023-02-18',
'boiled egg', 'breakfast', 1, 2),

('2023-02-18', 'apple', 'breakfast', 1, 3), ('2023-02-18', 'Rye
Bread', 'breakfast', 2, 3),
('2023-02-18', 'coffee with milk without sugar', 'breakfast', 2, 3),

('2023-02-18', 'Pear', 'breakfast', 1, 4), ('2023-02-18', 'Whole
wheat bread', 'breakfast', 1, 4),
('2023-02-18', 'Milk lowfat', 'breakfast', 1, 4), ('2023-02-18',
'banana', 'breakfast', 1, 4),

('2023-02-18', 'Vegetable kofta curry', 'lunch', 1, 1), ('2023-02-18',
'Boiled rice', 'lunch', 1, 1),
('2023-02-18', 'Whole wheat bread', 'lunch', 2, 1), ('2023-02-18',
'Potato-beans mixed vegetable', 'lunch', 1, 1),
('2023-02-18', 'coffee with milk without sugar', 'lunch', 1, 1),

('2023-02-18', 'Whole grain tortilla', 'lunch', 1, 2), ('2023-02-18',
'Salmon', 'lunch', 2, 2),
('2023-02-18', 'Dal Makhani', 'lunch', 1, 2), ('2023-02-18', 'Boiled
rice', 'lunch', 1, 2),
('2023-02-18', 'coffee with milk without sugar', 'lunch', 1, 2),

('2023-02-18', 'Quinoa cooked', 'lunch', 1, 3), ('2023-02-18', 'Whole
wheat bread', 'lunch', 2, 3),
('2023-02-18', 'Potato-baigan mixed vegetable', 'lunch', 2, 3),
('2023-02-18', 'coffee black without sugar', 'lunch', 1, 3),

('2023-02-18', 'Chicken breast', 'lunch', 2, 4), ('2023-02-18',
'Whole grain tortilla', 'lunch', 1, 4),
('2023-02-18', 'Boiled rice', 'lunch', 1, 4), ('2023-02-18', 'Salmon',
'lunch', 1, 4),
('2023-02-18', 'coffee black without sugar', 'lunch', 1, 4),

('2023-02-18', 'Pear', 'snacks', 1, 1), ('2023-02-18', 'dark
chocolate', 'snacks', 1, 1),
('2023-02-18', 'biscuits', 'snacks', 2, 1),

('2023-02-18', 'blueberries', 'snacks', 1, 2), ('2023-02-18', 'Yogurt
nonfat', 'snacks', 1, 2),
('2023-02-18', 'coffee black without sugar', 'snacks', 1, 2),

('2023-02-18', 'apple', 'snacks', 1, 3), ('2023-02-18', 'Rye Bread',
'snacks', 1, 3),
('2023-02-18', 'coffee with milk without sugar', 'snacks', 1, 3),

('2023-02-18', 'Pear', 'snacks', 1, 4), ('2023-02-18', 'Popcorn',
'snacks', 1, 4),
('2023-02-18', 'coffee with milk without sugar', 'snacks', 1, 4),
('2023-02-18', 'biscuits', 'snacks', 2, 4),

('2023-02-18', 'Chicken breast', 'dinner', 1, 1), ('2023-02-18',
'vegetable soup', 'dinner', 1, 1),
('2023-02-18', 'icecream', 'snacks', 1, 1),
```

('2023-02-18', 'Chicken breast', 'dinner', 1, 2), ('2023-02-18', 'Whole wheat bread', 'dinner', 1, 2),  
('2023-02-18', 'Lentil soup', 'dinner', 1, 2),  
  
('2023-02-18', 'Whole grain tortilla', 'dinner', 2, 3), ('2023-02-18', 'Potato-baigan mixed vegetable', 'dinner', 1, 3),  
('2023-02-18', 'apple', 'dinner', 1, 3),  
  
('2023-02-18', 'Rye Bread', 'dinner', 1, 4), ('2023-02-18', 'grilled fish', 'dinner', 2, 4),  
('2023-02-18', 'Pear', 'dinner', 1, 4),  
  
('2023-02-19', 'Pear', 'breakfast', 1, 1), ('2023-02-19', 'Whole wheat bread', 'breakfast', 2, 1),  
('2023-02-19', 'boiled egg', 'breakfast', 1, 1), ('2023-02-19', 'coffee black without sugar', 'breakfast', 1, 1),  
  
('2023-02-19', 'apple', 'breakfast', 1, 2), ('2023-02-19', 'Whole wheat bread', 'breakfast', 1, 2),  
('2023-02-19', 'Milk lowfat', 'breakfast', 1, 2), ('2023-02-18', 'banana', 'breakfast', 1, 2),  
  
('2023-02-19', 'boiled egg', 'breakfast', 2, 3), ('2023-02-19', 'Rye Bread', 'breakfast', 1, 3),  
('2023-02-19', 'coffee black without sugar', 'breakfast', 1, 3),  
  
('2023-02-19', 'Pear', 'breakfast', 2, 4), ('2023-02-19', 'Whole wheat bread', 'breakfast', 1, 4),  
('2023-02-19', 'Milk lowfat', 'breakfast', 1, 4),  
  
('2023-02-19', 'Vegetable kofta curry', 'lunch', 1, 1), ('2023-02-19', 'Boiled rice', 'lunch', 1, 1),  
('2023-02-19', 'boiled egg', 'lunch', 1, 1), ('2023-02-19', 'coffee black without sugar', 'lunch', 1, 1),  
  
('2023-02-19', 'Whole grain tortilla', 'lunch', 1, 2), ('2023-02-19', 'Quinoa cooked', 'lunch', 1, 2),  
('2023-02-19', 'Chickpeas', 'lunch', 1, 2), ('2023-02-19', 'grilled fish', 'lunch', 1, 2),  
('2023-02-19', 'coffee with milk without sugar', 'lunch', 1, 2),  
  
('2023-02-19', 'Quinoa cooked', 'lunch', 1, 3), ('2023-02-19', 'Rye Bread', 'lunch', 2, 3),  
('2023-02-19', 'Dal Makhani', 'lunch', 1, 3), ('2023-02-19', 'grilled fish', 'lunch', 1, 3),  
('2023-02-19', 'coffee black without sugar', 'lunch', 1, 3),  
  
('2023-02-19', 'Chicken breast', 'lunch', 2, 4), ('2023-02-19', 'Whole grain tortilla', 'lunch', 1, 4),  
('2023-02-19', 'Quinoa cooked', 'lunch', 1, 4), ('2023-02-19', 'Dal Makhani', 'lunch', 1, 4),  
('2023-02-19', 'coffee black without sugar', 'lunch', 1, 4),  
  
('2023-02-19', 'Pear', 'snacks', 1, 1), ('2023-02-19', 'Rye Bread', 'snacks', 1, 1),  
  
('2023-02-19', 'Watermelon diced', 'snacks', 1, 2), ('2023-02-19', 'Yogurt nonfat', 'snacks', 1, 2),



```

('2023-02-19', 'apple', 'snacks', 1, 3), ('2023-02-19', 'coffee black
without sugar', 'snacks', 1, 3),
('2023-02-19', 'biscuits', 'snacks', 2, 3),

('2023-02-19', 'Pear', 'snacks', 1, 4), ('2023-02-19', 'Milk lowfat',
'snacks', 1, 4),

('2023-02-19', 'chicken pizza', 'dinner', 3, 1), ('2023-02-19', 'soft
drink', 'dinner', 1, 1),
('2023-02-19', 'coffee black without sugar', 'dinner', 1, 1),

('2023-02-19', 'chicken burger', 'dinner', 1, 2), ('2023-02-19',
'french fries', 'dinner', 1, 2),
('2023-02-19', 'soft drink', 'dinner', 1, 2),

('2023-02-19', 'grilled fish', 'dinner', 1, 3), ('2023-02-19', 'Rye
Bread', 'dinner', 1, 3),
('2023-02-19', 'Lentil soup', 'dinner', 1, 3),

('2023-02-19', 'Salmon', 'dinner', 1, 4), ('2023-02-19', 'Whole wheat
bread', 'dinner', 1, 4),
('2023-02-19', 'dark chocolate', 'dinner', 1, 4), ('2023-02-19',
'coffee black without sugar', 'dinner', 1, 4)
;

```

**SQL queries annotated that demonstrate the scope, potential and functionality of database.**

**1. The mentioned SQL Query gives the result about the Food items that have less than 20gms protein and more than 5gms of fat**

```

Select FoodName, TotalProtein_gms, TotalFat_gms
from macronutrients where TotalProtein_gms < 20 and TotalFat_gms > 5;

```

```

> Select FoodName, TotalProtein_gms, TotalFat_gms
from macronutrients where TotalProtein_gms < 20 and TotalFat_gms > 5

```

FoodName	TotalProtein_gms	TotalFat_gms
biscuits	3.2	8.5
boiled egg	6.3	5.3
dark chocolate	2	12
french fries	1.9	13.1
icecream	3.52	10.72
Potato-beans		
mixed vegetable	2.4	8.7
Salmon	17	5.4
Vegetable		
kofta curry	2.6	9.1
Yogurt nonfat	11	5.8
NULL	NULL	NULL

10 rows

2. The mentioned SQL Query gives Maximum calories that a food item has in Macronutrients Table.

```
Select FoodName, Serving_size, Calories_Kcal from macronutrients
where calories_Kcal =( Select max(calories_kcal) from macronutrients);
```

```
> Select FoodName, Serving_size, Calories_Kcal from macronutrients
where calories_Kcal =( Select max(calories_kcal) from macronutrients)
```

+	-----	+	-----	+	-----	+
	FoodName		Serving_size		Calories_Kcal	
+	-----	+	-----	+	-----	+
	chicken burger		1		535	
	NULL		NULL		NULL	
+	-----	+	-----	+	-----	+

2 rows

### 3. SUB QUERY:

The mentioned SQL Query gives Print ID, Name, Age of the user who was FoodLogID 110.

```
SELECT UserID, UserName, Age from User
where User.UserID = (SELECT FoodLog.UserID FROM FoodLog
WHERE FoodLogID = 110 GROUP BY UserID);
```

```
> SELECT UserID, UserName, Age from User
where User.UserID = (SELECT FoodLog.UserID FROM FoodLog
WHERE FoodLogID = 110 GROUP BY UserID)
```

+	-----	+	-----	+	-----	+
	UserID		UserName		Age	
+	-----	+	-----	+	-----	+
	3		Lokesh		35	
	NULL		NULL		NULL	
+	-----	+	-----	+	-----	+

2 rows

4. The mentioned SQL Query gives the total calories for each FoodLogID and print the FoodLogID and TotalCalories with FoodLogDate 18/02/2023 in breakfast and order FoodLogID in Ascending order .

```
SELECT fl.FoodLogID, mn.Calories_Kcal * fl.Quantity AS TotalCalories
FROM FoodLog fl, MacroNutrients mn
where fl.FoodName = mn.FoodName
AND fl.MealTime = 'breakfast'
AND FL.FoodlogDate = '2023-02-18'
ORDER BY FoodLogID;
```

```
> SELECT fl.FoodLogID, mn.Calories_Kcal * fl.Quantity AS TotalCalories
FROM FoodLog fl, MacroNutrients mn
where fl.FoodName = mn.FoodName
AND fl.MealTime = 'breakfast'
AND FL.FoodlogDate = '2023-02-18'
ORDER BY FoodLogID
```

FoodLogID	TotalCalories
101	81
102	112
103	77
104	86
105	82
106	56
107	121
108	77
109	81
110	112
111	12
112	98
113	56
114	121
115	105
167	105

16 rows

## 5. INNER JOIN QUERY:

The mentioned SQL Query gives FoodNames, MealTime and Calories for all the food logged by userID 2 on the FoodLogDate 19/02/2023 and order calories in descending order .

```
SELECT f.UserID, f.FoodName, F.MealTime, m.Calories_Kcal
from FoodLog f
inner join macronutrients m
on f.FoodName = m.FoodName where f.UserID = 2 and f.FoodLogDate =
'2023-02-19'
order by m.Calories_Kcal desc;
```

```
> SELECT f.UserID, f.FoodName, F.MealTime, m.Calories_Kcal
from FoodLog f
inner join macronutrients m
on f.FoodName = m.FoodName where f.UserID = 2 and f.FoodLogDate = '2023-02-19'
order by m.Calories_Kcal desc
```

UserID	FoodName	MealTime	Calories_Kcal
2	chicken burger	dinner	535
2	french fries	dinner	196
2	soft drink	dinner	140
2	Yogurt nonfat	snacks	127
2	Whole grain tortilla	lunch	127
2	Milk lowfat	breakfast	121
2	Quinoa cooked	lunch	115
2	grilled fish	lunch	109
2	apple	breakfast	81
2	Chickpeas	lunch	74
2	Whole wheat bread	breakfast	56
2	Watermelon diced	snacks	50
2	coffee with milk without sugar	lunch	6

13 rows

## 6. VIEW :

The mentioned SQL Query gives FoodLogID, UserNames, FoodNames of all the food items that were logged more than 25g of Carbohydrates. Save as a view named HighCarbsDiet .

```
Create View HighCarbsDiet AS SELECT u.UserName, fl.FoodLogID,
fl.FoodName, mn.TotalCarbs_gms
FROM User u, FoodLog fl, MacroNutrients mn
where u.UserID = fl.UserID
AND fl.FoodName = mn.FoodName
AND TotalCarbs_gms >= 25;
```

```
select * from HighCarbsDiet;
```

UserName	FoodLogID	FoodName	TotalCarbs_gms
Lalitha	117	Boiled rice	25
Lalitha	119	Potato-beans mixed vegetable	34.3
Lalitha	135	Pear	25.1
Lalitha	136	dark chocolate	25
Lalitha	160	Pear	25.1
Lalitha	175	Boiled rice	25
Lalitha	193	Pear	25.1
Lalitha	202	chicken pizza	25.47
Lalitha	203	soft drink	36.09
Sureka	124	Boiled rice	25
Sureka	153	Lentil soup	26.61
Sureka	167	banana	26.7
Sureka	205	chicken burger	57
Sureka	207	soft drink	36.09
Lokesh	128	Potato-baigan mixed vegetable	32.3
Lokesh	155	Potato-baigan mixed vegetable	32.3
Lokesh	210	Lentil soup	26.61
Karthik	112	Pear	25.1
Karthik	115	banana	26.7
Karthik	132	Boiled rice	25
Karthik	144	Pear	25.1
Karthik	159	Pear	25.1
Karthik	171	Pear	25.1
Karthik	200	Pear	25.1
Karthik	213	dark chocolate	25

25 rows

## 7. TRIGGERS:

The mentioned SQL Query tell about how to Create Trigger in the database such that exercises with the ExeCountCalories\_Kcal 0 cannot be added, when tried to insert the value with the INSERT INTO, it has to show the invalid value for the column.

```
DELIMITER //
```

```
CREATE TRIGGER trigger1
    before insert on Exercise for
    each row
begin
    IF (NEW.ExeCountCalories_Kcal <= 0) THEN
        CALL `Error: Invalid value for column read New record`;
    end IF;
END //
```

```
delimiter ;
```

```
INSERT INTO Exercise (Date, Category, ExeCountCalories_kcal, UserID)
VALUES ('2023-02-20', 'swimming', 0, 3);
```

The result shown below when I tried to insert values in exercise table

```
23:35:03    INSERT INTO Exercise (Date, Category,
ExeCountCalories_kcal, UserID) VALUES ('2023-02-20', 'swimming', 0,
3)    Error Code: 1305. PROCEDURE nutrition_db.Error: Invalid value
for column read New record does not exist 0.000 sec
|
```

## 8. Stored Procedure :

The mentioned SQL Query gives information about how to Calculate total calories intake by mealTime, FoodLogDate, and UserID using stored procedure and also Called the procedure.

```
CREATE PROCEDURE calculate_calories_by_meal_type(  
    IN Foodlog_Date DATE,  
    IN Meal_Time VARCHAR(32),  
    IN User_ID INT,  
    OUT totalcalories INT  
)  
BEGIN  
    SELECT SUM(mn.Calories_Kcal * fl.Quantity) AS total_calories  
    FROM FoodLog fl  
    INNER JOIN MacroNutrients mn ON fl.FoodName = mn.FoodName  
    WHERE MealTime = Meal_Time AND FoodLogDate = FoodLog_date  
    AND UserID = User_ID  
    GROUP BY MealTime, UserID;  
  
    END //
```

delimiter ;

```
CALL calculate_calories_by_meal_type('2023-02-18', 'snacks', '2',  
@totalcalories);  
> CALL calculate_calories_by_meal_type('2023-02-18', 'snacks', '2', @totalcalories)
```

total_calories
212

1 rows

---

### Use of time

The final amount of work approximates around 1 week (around 2 hours per day).

### Requirements definition and concept modeling

The conceptual model took me time and I thought two models at last I finalized the mentioned one. It took about 2 hours.

### ER CHART and database creation

During the creation phase of the ER diagram, I thought about the contents of the boards and their relationships. It took long time for forward engineering. At this stage the type of Data type to be mentioned for quantities and serving size in the boards was little bit confused but the end it resolved quickly by observing example databases in the google.

### Data addition and queries



Adding data to the boards was easy, but I had to do it again a few times. The first time I forgotten to enter few contents in the board before forward engineer. I made changes to the tables at this point I saved the insert queries which helped me not to enter twice. Collecting the data information like exact macronutrients of each product from different websites related to their calories per serving , calories etc and also . It took around 6 hours for both collecting information and inserting them in the tables.

Different queries are tried on this nutrition database to check the functionality.

- To find the maximum calories the fooditem has in macronutrients.
- Finding Fooditems using where operators(> or <).
- Find the user details based on the foodlogId using subquery.
- To find the total calories for each foodlogId for particular foodlogdate and mealtime
- To find foodnames,mealtime,calories for all the foodlogged by user on particular foodlogdate. (Using innerjoin query)
- Created a view HighCarbsDiet which includes foodlogid,user, foodname of all fooditems that are logged more than 25gms of carbs.
- Trigger concept on the table exercises with the 0 calories cannot be added.
- Using stored Procedures the total calories intake by mealtime,foodlogdate and userid.

The total time spend at this stage is more than the other stages majorly on stored procedure various examples model are observed and learnt this concept and later tried this concept on this nutrition database. At first I got wrong results , I cross checked every step and corrected the mistake the naming in statements in the create procedure block was different with the where condition. At last i got the results. It took 8 hours.

### **Reportage**

For reporting, I took less time , at every stage I saved every thing in my documentation including the snipping images of the different stages of the work and organized them for later use. The report part took about 2 hours.

### **Self-assessments:**

At early stages I thought it was tough to finish this practice task by single. When I finished each part of the task step by step I got confidence that I can do it. However, I learned a lot while doing the practice task . I am not expecting excellent grade but I tried as much as I learned from this course to present and report this work. In my opinion, ms access prototype is some what confusing , so I didn't start and the information was not clear. The assessment of my own competence in relation to the practice work is 3.5.