Group 6: Bibek Manandhar, Sachin Karki

Dr. Omar Darwish

Database Management, CSCI 3423

21st November, 2020

Final Project Implementation - Nutrition Tracking Database

1. Project Goals

As mentioned, and discussed previously health is a major aspect of our life. Health is what helps us to determine the entirety of our well-beingness and depicts the state of human anatomy in terms of physical, mental and emotional aspects. In order to ensure sound health, one needs to monitor eating habits and physical activities to make sure they are on good track. An easy way to manage and monitor those habits and activities is by using a proper record keeping system. The purpose and goal of this project is to become that proper record keeping system. Nutrition Tracking database is capable of recording information that can help one to understand the pattern of their health progress. Not only it will keep the record of their diet and exercise, it will aware people about the state of their health.

2. Database Description

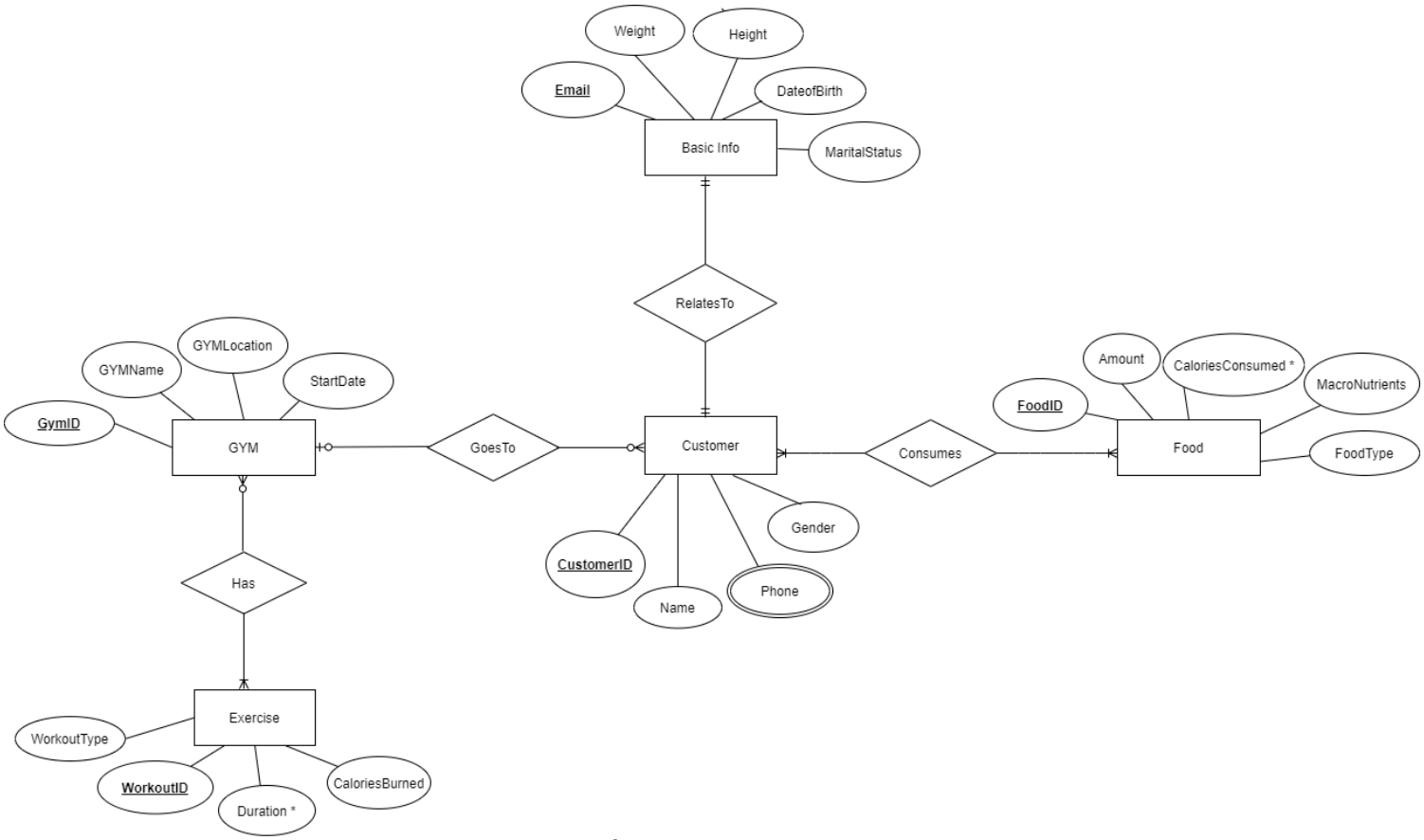
Nutrition Tracking Database does exactly what it sounds like. The database consists of several information and class to make it possible to record the data. For an instance, the database relies on three sets of account to track a data. These three sets are Customer Information, Food and Exercise. However, these three sets can be expanded into multiple sets of additional accounts to make it more relevant and accurate. To begin with, customer information will mostly store

general information of a customer like their name, gender, date of birth, height and weight. In addition to this, customer ID, marital status and contact information adds up to make that part more solid. Food is another entity that specializes in recording what type of food they ate, what was the amount of the food they ate, how much kilo calories did they gain. Lastly, exercise is another aspectual entity that takes advantage in recording what sort of activities did they perform whether it is cardio, stretch or lift. For how long did they work out and how much kilo calories did they burn. Typically, 120 kilo calories are burned by 30 minutes of walking and somewhere between 150-170 kilo calories is burned by an intense workout session.

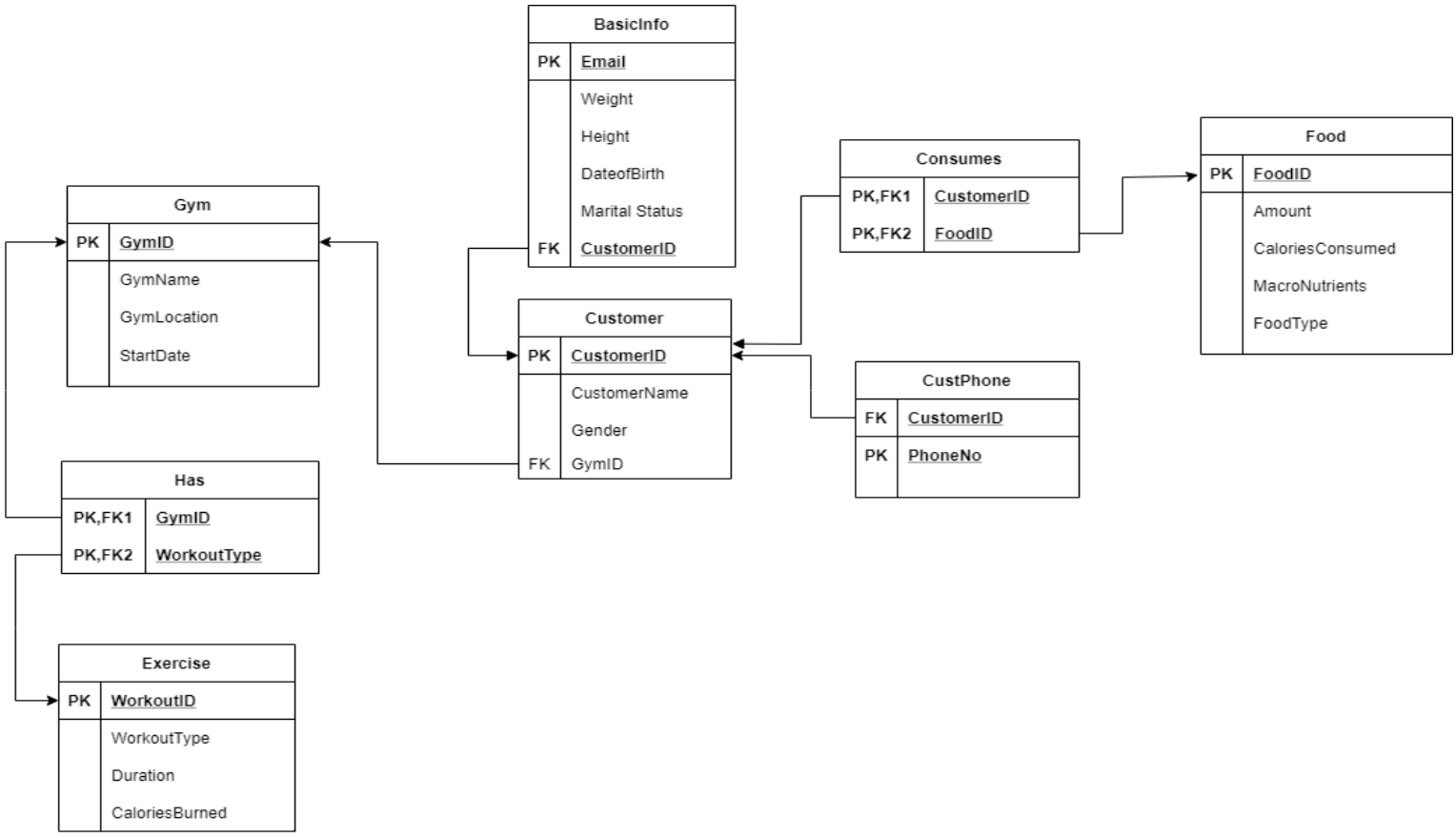
This information will pave a way for us to know exactly what kind of workout will help us burn how much of calories. Users will benefit in lot of ways with an effective and consistent usage of the database. With an accurate and precise set of information they feed, users will be able to keep track of overall calorie consumption. Every food has a different calorie count, this would allow them to be concise about which food to take and which to avoid. Users are also able to realize the right set of workouts for them. With regular update on calorie gain and calorie loss, user will have the insights on calorie ration. To sum up, users will benefit the privilege of being able to calculate BMI (Body Mass Index) easily based on these records and calculation.

3. Data Model & Design

a. ER Diagrams: For ER diagrams, all the resulting and relating sketches were done in draw.io website. After creative thinking and drafting a rough version of the database, a conceptual model was envisioned which was later on improvised and improved with the table and columns that were created early in the process. The ER was further developed into a meaningful relational schema that allowed us to shape and code the entirety of the database we created later on. Both ER and Relational Schema are pictured below.

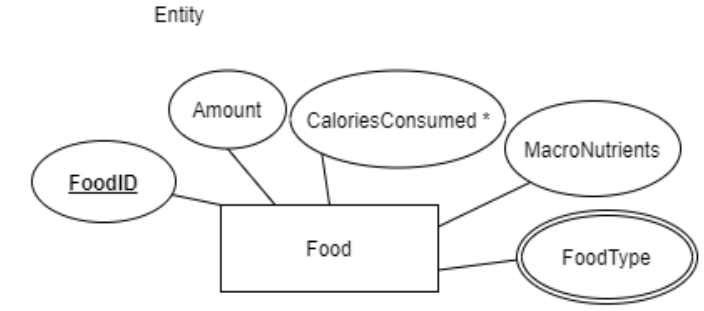


ER Diagram of the Nutrition Tracking Database

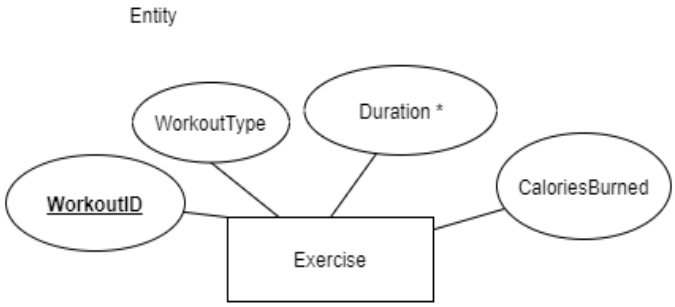


Relational Schema of Nutrition Tracking Database

b. Business Rules: A couple of business rules were added in the database to keep it more relevant and accurate. One of the tables that was created in the development phase demanded an adjustment for different gender. In the food entity, it was realized that male and female have different requirement for daily calorie intake. Male typically needs a 2500 kcal per day while female needs 2000 kcal per day. Therefore, a business rule was added in calorie consumption. Similarly, duration of workout also needed an adjustment. To qualify an activity as a 'workout' a minimum duration of that activity performed is needed. That being realized a minimum of 45 minutes of workout is a must. That way it was possible for us to add another business rule in duration of exercise entity. These two business rules are pictured below.



* Calorie Consumption Must be a Minimum of 2000 for Women and 2500 for Men



* Duration of a workout must be 45 minutes minimum and 2 hours maximum

Resulting Relation

Food	
PK	FoodID
	Amount
	CaloriesConsumed
	MacroNutrients

Business Rule for Calorie Consumption

Resulting Relation

Exercise	
PK	WorkoutID
	WorkoutType
	Duration
	CalorieBurned

Business Rule for Duration of Workout

c. ***Data Dictionary***: A quick way to access the information from the database is by using a data dictionary. To overview and sum up major aspects of the database, listed below is the data dictionary of the Nutrition Tracking Database.

TableName	**ColumnDescription**	**DataTypes**	**ColumnRestrictions**
BasicInfo	Email	VARCHAR	50
BasicInfo	Weight	VARCHAR	10
BasicInfo	Height	VARCHAR	10
BasicInfo	DateofBirth	DATE	DATE
BasicInfo	MaritalStatus	CHAR	15
Customer	CustomerID	CHAR	3
Customer	CustomerName	VARCHAR	40
Customer	Gender	CHAR	10
CustPhone	PhoneNumber	VARCHAR	12
CustPhone	CustomerID	CHAR	3
Food	FoodID	VARCHAR	3
Food	FoodType	CHAR	15
Food	Amount	FLOAT	15
Food	CalorieConsumed	FLOAT	20
Food	MacroNutrients	CHAR	20
Consumes	CustomerID	CHAR	3
Consumes	FoodID	VARCHAR	3
Gym	GymID	CHAR	5

Gym	GymName	VARCHAR	30
Gym	GymLocation	VARCHAR	30
Gym	StartDate	DATE	DATE
Gym	CustomerID	CHAR	3
Exercise	WorkoutID	VARCHAR	3
Exercise	WorkoutType	CHAR	40
Exercise	Duration	VARCHAR	20
Exercise	CaloriesBurned	FLOAT	15
Has	GymID	CHAR	5
Has	WorkoutID	VARCHAR	3

4. Implementation

a. Creating Database in MySQL Server and MySQL Workbench: After creating entity relation diagram and sketching relational schema for the database we used MySQL Workbench to carefully code each tables and columns. The resulting code is down below.

SQL Code:

```
Create schema Project;
Use Project;

CREATE TABLE gym (
   gymid              CHAR(5)            NOT NULL,
   gymname            VARCHAR(30)        NOT NULL,
   gymlocation        VARCHAR(30)        NOT NULL,
   startdate          DATE               NOT NULL,
   PRIMARY KEY (gymid)
);
CREATE TABLE customer (
   customerid         CHAR(3)            NOT NULL,
   customername       VARCHAR(40)        NOT NULL,
   gender             CHAR(10)           NOT NULL,
   gymid CHAR(5) NOT NULL,
   PRIMARY KEY (customerid),
   FOREIGN KEY (gymid) REFERENCES gym (gymid)
);
CREATE TABLE basicinfo (
   email              VARCHAR(50)        NOT NULL,
   weight             VARCHAR(10)        NOT NULL,
   height             VARCHAR(10)        NOT NULL,
   dateofbirth        DATE               NOT NULL,
   maritalstatus      CHAR(15)           NOT NULL,
   customerid         CHAR(3)            NOT NULL,
   PRIMARY KEY (email),
   FOREIGN KEY (customerid) REFERENCES customer (customerid)
);
CREATE TABLE custphone (
   phonenumber        VARCHAR(12)        NOT NULL,
   customerid         CHAR(3)            NOT NULL,
   FOREIGN KEY (customerid) REFERENCES customer (customerid)
);
CREATE TABLE food (
   foodid             VARCHAR(3)         NOT NULL,
   foodtype           CHAR(15)           NOT NULL,
   amount             FLOAT              NOT NULL,
   caloriesconsumed   FLOAT              NOT NULL,
   macronutrients     CHAR(20)           NOT NULL,
   PRIMARY KEY (foodid)
);
CREATE TABLE consumes (
   customerid         CHAR(3)            NOT NULL,
   foodid             VARCHAR(3)         NOT NULL,
   FOREIGN KEY (customerid) REFERENCES customer (customerid),
   FOREIGN KEY (foodid)  REFERENCES food (foodid)
);
CREATE TABLE exercise (
   workoutid          VARCHAR(3)         NOT NULL,
   workouttype        CHAR(40)           NOT NULL,
   duration           VARCHAR(20)        NOT NULL,
   caloriesburned     FLOAT              NOT NULL,
   PRIMARY KEY (workoutid)
);
CREATE TABLE has (
   gymid              CHAR(5)            NOT NULL,
   workoutid          VARCHAR(3)         NOT NULL,
   FOREIGN KEY (gymid) REFERENCES gym (gymid),
   FOREIGN KEY (workoutid) REFERENCES exercise (workoutid)
);
```

b. Populating the Database: After coding the required tables and columns, the next step was to feed records and information into the database. In order to populate, 20 entries of records were given into the database. The resulting code is given below.

i. Populating the customer column

```
INSERT INTO customer VALUES ('101','Ted Marquis','Male','34812');
INSERT INTO customer VALUES ('102','Sandy Yule','Male','20318');
INSERT INTO customer VALUES ('103','Rosendo Hefner','Male','39028');
INSERT INTO customer VALUES ('104','Quentin Tibbits','Male','28730');
INSERT INTO customer VALUES ('105','Jonathan Thiem','Male','27364');
INSERT INTO customer VALUES ('106','Clayton Parekh','Male','17283');
INSERT INTO customer VALUES ('107','Christoper Lofland','Male','26354');
INSERT INTO customer VALUES ('108','Alejandro Kubiak','Male','12093');
INSERT INTO customer VALUES ('109','Carlos Albino','Male','28374');
INSERT INTO customer VALUES ('110','Cruz Burtch','Male','29837');
INSERT INTO customer VALUES ('111','Annmarie Whetstone','Female','26743');
INSERT INTO customer VALUES ('112','Vannesa Uresti','Female','24534');
INSERT INTO customer VALUES ('113','Olene Greco','Female','26712');
INSERT INTO customer VALUES ('114','Cyndy Sells','Female','19023');
INSERT INTO customer VALUES ('115','Yajaira Oyler','Female','10293');
INSERT INTO customer VALUES ('116','Jennette Tolman','Female','12283');
INSERT INTO customer VALUES ('117','Antonina Magee','Female','28839');
INSERT INTO customer VALUES ('118','Rose Zerby','Female','18297');
INSERT INTO customer VALUES ('119','Delphia Galang','Female',21723');
INSERT INTO customer VALUES ('120','Alona Kubota','Female','12834');
```

ii. Populating the basicinfo column

```
INSERT INTO basicinfo VALUES ('tedmarquis@twu.edu', '101 kg' ,'173 cm', '1975/12/7', 'Married', '101');
INSERT INTO basicinfo VALUES ('sandyyule@twu.edu', '120 kg' ,'149 cm', '1974/01/24', 'Unmarried', '102');
INSERT INTO basicinfo VALUES ('rosendohefner@twu.edu', '100 kg' ,'160 cm', '1992/12/17', 'Married', '103');
INSERT INTO basicinfo VALUES ('quentintibbitis@twu.edu', '95 kg' ,'153 cm', '1990/12/27', 'Married', '104');
INSERT INTO basicinfo VALUES ('jonathanthiem@twu.edu', '79 kg' ,'123 cm', '1975/05/13', 'Unmarried', '105');
INSERT INTO basicinfo VALUES ('claytonparekh@twu.edu', '75 kg' ,'119 cm', '1971/09/27', 'Married', '106');
INSERT INTO basicinfo VALUES ('christoperlofland@gmail.com', '72 kg' ,'113 cm', '1973/10/14', 'Unmarried', '107');
INSERT INTO basicinfo VALUES ('alejandrokubiak@gmail.com', '97 kg' ,'134 cm', '1982/09/20', 'unmarried', '108');
INSERT INTO basicinfo VALUES ('carlosalbina@rocketmail.com', '95 kg' ,'145 cm', '1988/06/27', 'Married', '109');
INSERT INTO basicinfo VALUES ('cruzburtch@yahoo.com', '65 kg' ,'151 cm', '1974/12/23', 'Unmarried', '110');
INSERT INTO basicinfo VALUES ('annamariewhetsone@gmail.com', '99 kg' ,'155 cm', '1993/11/26', 'Married', '111');
INSERT INTO basicinfo VALUES ('vannesauresti@gmail.com', '75 kg' ,'167 cm', '1982/03/07', 'Married', '112');
INSERT INTO basicinfo VALUES ('olenegreco@rocketmail.com', '150 kg' ,'179 cm', '1984/01/05', 'Unmarried', '113');
INSERT INTO basicinfo VALUES ('cyndysells@gmail.com', '110 kg' ,'164 cm', '1999/12/24', 'Married', '114');
INSERT INTO basicinfo VALUES ('yajairaoyler@rocketmail.com', '90 kg' ,'132 cm', '1972/05/08', 'Unmarried', '115');
INSERT INTO basicinfo VALUES ('jennettetolman@yahoo.com', '80 kg' ,'136 cm', '1992/10/06', 'Married', '116');
INSERT INTO basicinfo VALUES ('antoninamagee@yahoo.com', '71 kg' ,'167 cm', '1997/03/08', 'Unmarried', '117');
INSERT INTO basicinfo VALUES ('rosezerby@gmail.com', '55 kg' ,'148 cm', '1981/09/26', 'Unmarried', '118');
INSERT INTO basicinfo VALUES ('delphiagalang@twu.edu', '65 kg' ,'131 cm', '1980/11/19', 'Married', '119');
INSERT INTO basicinfo VALUES ('alonakubota@gmail.com', '110 kg' ,'124 cm', '2000/06/05', 'Unmarried', '120');
```

iii. Populating into custphone column

```
INSERT INTO custphone VALUES ('301-559-5574','101');
INSERT INTO custphone VALUES ('786-973-5150','102');
INSERT INTO custphone VALUES ('804-720-9568','103');
INSERT INTO custphone VALUES ('804-755-2917','103');
INSERT INTO custphone VALUES ('660-256-0696','104');
INSERT INTO custphone VALUES ('305-608-3417','105');
INSERT INTO custphone VALUES ('815-867-0514','105');
INSERT INTO custphone VALUES ('704-328-0960','106');
INSERT INTO custphone VALUES ('602-819-1326','106');
INSERT INTO custphone VALUES ('773-972-6969','106');
INSERT INTO custphone VALUES ('415-785-9286','107');
INSERT INTO custphone VALUES ('480-610-5150','108');
INSERT INTO custphone VALUES ('917-470-4749','109');
INSERT INTO custphone VALUES ('323-554-0534','110');
INSERT INTO custphone VALUES ('817-703-8762','110');
INSERT INTO custphone VALUES ('406-513-2388','111');
INSERT INTO custphone VALUES ('989-501-1080','112');
INSERT INTO custphone VALUES ('806-776-6909','113');
INSERT INTO custphone VALUES ('206-666-3820','113');
INSERT INTO custphone VALUES ('734-549-0888','114');
INSERT INTO custphone VALUES ('302-788-7604','114');
INSERT INTO custphone VALUES ('941-348-9673','115');
INSERT INTO custphone VALUES ('907-430-4062','116');
INSERT INTO custphone VALUES ('704-453-1826','117');
INSERT INTO custphone VALUES ('347-669-2153','118');
INSERT INTO custphone VALUES ('435-239-8387','119');
INSERT INTO custphone VALUES ('301-387-8675','120');
```

iv. Populating into food

```
INSERT INTO food VALUES ('F1','Vegetarian',1809, 2100, 'Protein' );
INSERT INTO food VALUES ('F2','Vegan',1720,1980, 'Fiber');
INSERT INTO food VALUES ('F3','Low Carbs',1600, 1990, 'Carbs' );
INSERT INTO food VALUES ('F4','Plant Based',2000, 2400, 'Protein' );
INSERT INTO food VALUES ('F5','Zone Diet',1700, 2600, 'Fiber' );
INSERT INTO food VALUES ('F6','Ketogenic',1660, 2700, 'Carbs' );
INSERT INTO food VALUES ('F7','Meat Based',1570,3200, 'Protein' );
INSERT INTO food VALUES ('F8','Zone Diet',1293, 2300, 'Carbs' );
INSERT INTO food VALUES ('F9','Vegetarian',1578, 2900, 'Fiber' );
INSERT INTO food VALUES ('F10','Low Carbs',1890, 2500, 'Protein' );
INSERT INTO food VALUES ('F11','Plant Based',1782, 2900, 'Fiber' );
INSERT INTO food VALUES ('F12','Ketogenic',1453, 2199, 'Protein' );
INSERT INTO food VALUES ('F13','Vegan',1341, 3400, 'Carbs' );
INSERT INTO food VALUES ('F14','Plant Based',2131, 2300, 'Sugar' );
INSERT INTO food VALUES ('F15','Ketogenic',2100, 2400, 'Sugar' );
INSERT INTO food VALUES ('F16','Low Carbs',1400, 2560, 'Fiber' );
INSERT INTO food VALUES ('F17','Vegetarian',2000, 2100, 'Carbs' );
INSERT INTO food VALUES ('F18','Plant Based',1740, 2810, 'Protein' );
INSERT INTO food VALUES ('F19','Vegan',1400, 2710, 'Carbs' );
INSERT INTO food VALUES ('F20','Ketogenic',1274, 2680, 'Protein' );
```

v. Populating into consumes

```
INSERT INTO consumes VALUES ('101','F1');
INSERT INTO consumes VALUES ('102','F2');
INSERT INTO consumes VALUES ('103','F3');
INSERT INTO consumes VALUES ('104','F4');
INSERT INTO consumes VALUES ('105','F5');
INSERT INTO consumes VALUES ('106','F6');
INSERT INTO consumes VALUES ('107','F7');
INSERT INTO consumes VALUES ('108','F8');
INSERT INTO consumes VALUES ('109','F9');
INSERT INTO consumes VALUES ('110','F10');
INSERT INTO consumes VALUES ('111','F11');
INSERT INTO consumes VALUES ('112','F12');
INSERT INTO consumes VALUES ('113','F13');
INSERT INTO consumes VALUES ('114','F14');
INSERT INTO consumes VALUES ('115','F15');
INSERT INTO consumes VALUES ('116','F16');
INSERT INTO consumes VALUES ('117','F17');
INSERT INTO consumes VALUES ('118','F18');
INSERT INTO consumes VALUES ('119','F19');
INSERT INTO consumes VALUES ('120','F20');
```

vi. Populating into gym

```
INSERT INTO gym VALUES ('34812','LA Fitness', 'Grand paire', '2019/12/18');
INSERT INTO gym VALUES ('20318','Anytime Fitness', 'Irving', '2020/01/19');
INSERT INTO gym VALUES ('39028','Fitness Connection', 'Forthworth', '2020/02/13');
INSERT INTO gym VALUES ('28730','24 Hour Fitness', 'Dallas', '2019/09/18');
INSERT INTO gym VALUES ('27364','Planet Fitness', 'Denton', '2020/05/25');
INSERT INTO gym VALUES ('17283','Wellness Fitness', 'Hurst', '2020/05/26');
INSERT INTO gym VALUES ('26354','Abbott Fitness', 'Hurst', '2020/03/01');
INSERT INTO gym VALUES ('12093','LA Fitness', 'Grand paire', '2019/11/21');
INSERT INTO gym VALUES ('28374','Fitness Connection', 'Dallas', '2019/09/15');
INSERT INTO gym VALUES ('29837','LA Fitness', 'Grapevine', '2019/10/10');
INSERT INTO gym VALUES ('26743','24 Hour Fitness', 'Eueless', '2019/12/17');
INSERT INTO gym VALUES ('24534','Wellness Fitness', 'Bedfrod', '2020/05/18');
INSERT INTO gym VALUES ('26712','Anytime Fitness', 'Grand paire', '2020/09/10');
INSERT INTO gym VALUES ('19023','Fitness Connection', 'Dallas', '2020/07/20');
INSERT INTO gym VALUES ('10293','24 Hour Fitness', 'Grand paire', '2020/08/17');
INSERT INTO gym VALUES ('12283','Anytime Fitness', 'Grapevine', '2020/09/09');
INSERT INTO gym VALUES ('28839','Fitness Connection', 'Dallas', '2020/03/14');
INSERT INTO gym VALUES ('18297','Anytime Fitness', 'Irving', '2019/07/12');
INSERT INTO gym VALUES ('21723','24 Hour Fitness', 'Forthworth', '2019/04/12');
INSERT INTO gym VALUES ('12834','LA Fitness', 'Forthworth', '2019/09/12');
```

vii. Populating into exercise

```
INSERT INTO exercise VALUES ('W1','Cardio', '40 mins', 300 );
INSERT INTO exercise VALUES ('W2','Aerobic Exercise', '60 mins',220);
INSERT INTO exercise VALUES ('W3','Anaerobic Exercise', '45 mins', 320);
INSERT INTO exercise VALUES ('W4','Circuit', '50 mins', 320 );
INSERT INTO exercise VALUES ('W5','Compound Exercises', '1 hour 20 mins', 400);
INSERT INTO exercise VALUES ('W6','Cross-Training', '60 mins', 290);
INSERT INTO exercise VALUES ('W7','Interval Training', '1 hour 30 mins', 250);
INSERT INTO exercise VALUES ('W8',' Isometrics', '1 hour', 280);
INSERT INTO exercise VALUES ('W9','Plyometrics', '40 mins', 200);
INSERT INTO exercise VALUES ('W10','Steady-State Cardio', '35 mins', 110);
INSERT INTO exercise VALUES ('W11','Strength Training', '1 hour 30 mins', 270);
INSERT INTO exercise VALUES ('W12','Tabata', '30 mins', 120);
INSERT INTO exercise VALUES ('W13','HIIT', '50 mins', 300);
INSERT INTO exercise VALUES ('W14','Heart Rate Zones', '40 mins', 265);
INSERT INTO exercise VALUES ('W15','Functional Moves', '30 mins', 130);
INSERT INTO exercise VALUES ('W16','Stretching', '30 mins', 110);
INSERT INTO exercise VALUES ('W17','Foam Rolling', '60 mins', 275);
INSERT INTO exercise VALUES ('W18','Flexibility', '30 mins', 114);
INSERT INTO exercise VALUES ('W19','Balance Exercise', '60 mins', 250);
INSERT INTO exercise VALUES ('W20','Yoga', '45 mins',210);
```

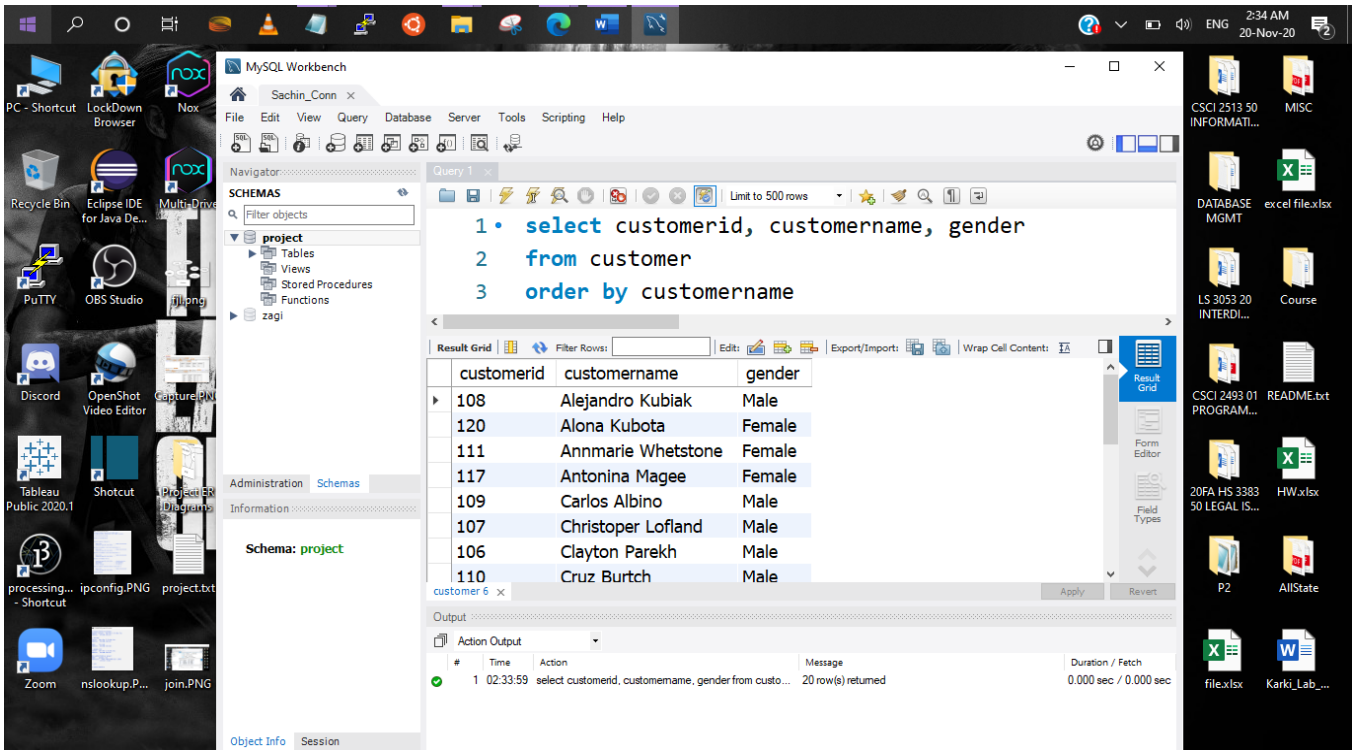
viii. Populating into has

```
INSERT INTO has VALUES ('34812','W1');
INSERT INTO has VALUES ('20318','W2');
INSERT INTO has VALUES ('39028','W3');
INSERT INTO has VALUES ('28730 ','W4');
INSERT INTO has VALUES ('27364','W5');
INSERT INTO has VALUES ('17283','W6');
INSERT INTO has VALUES ('26354','W7');
INSERT INTO has VALUES ('12093','W8');
INSERT INTO has VALUES ('28374','W9');
INSERT INTO has VALUES ('29837','W10');
INSERT INTO has VALUES ('26743','W11');
INSERT INTO has VALUES ('24534','W12');
INSERT INTO has VALUES ('26712','W13');
INSERT INTO has VALUES ('19023','W14');
INSERT INTO has VALUES ('10293','W15');
INSERT INTO has VALUES ('12283','W16');
INSERT INTO has VALUES ('28839','W17');
INSERT INTO has VALUES ('18297','W18');
INSERT INTO has VALUES ('21723','W19');
INSERT INTO has VALUES ('12834','W20');
```

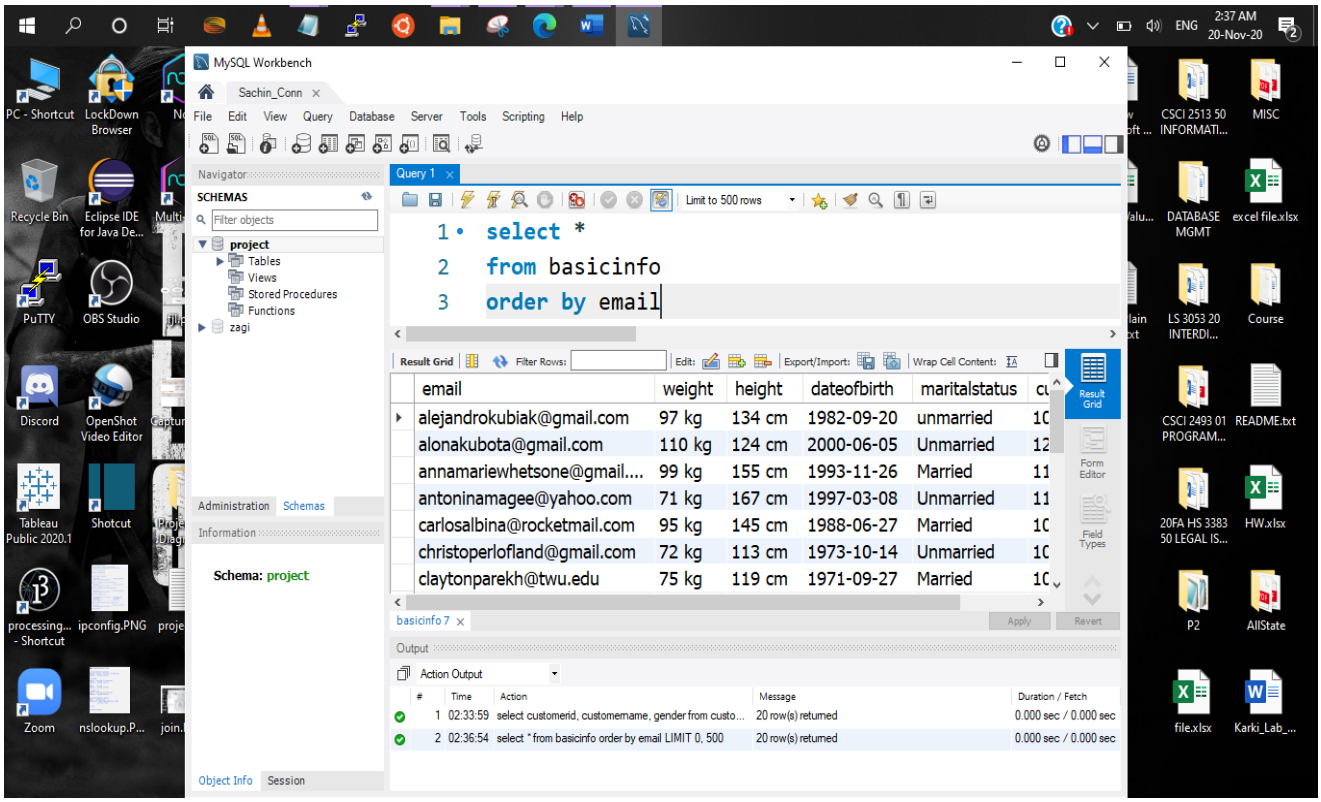
c. SQL Queries: Pictured below are some of the queries along with their results.

2 Trivial Query - simple select with ordering

Query 1: Retrieve the entire contents of the customer and order by customername

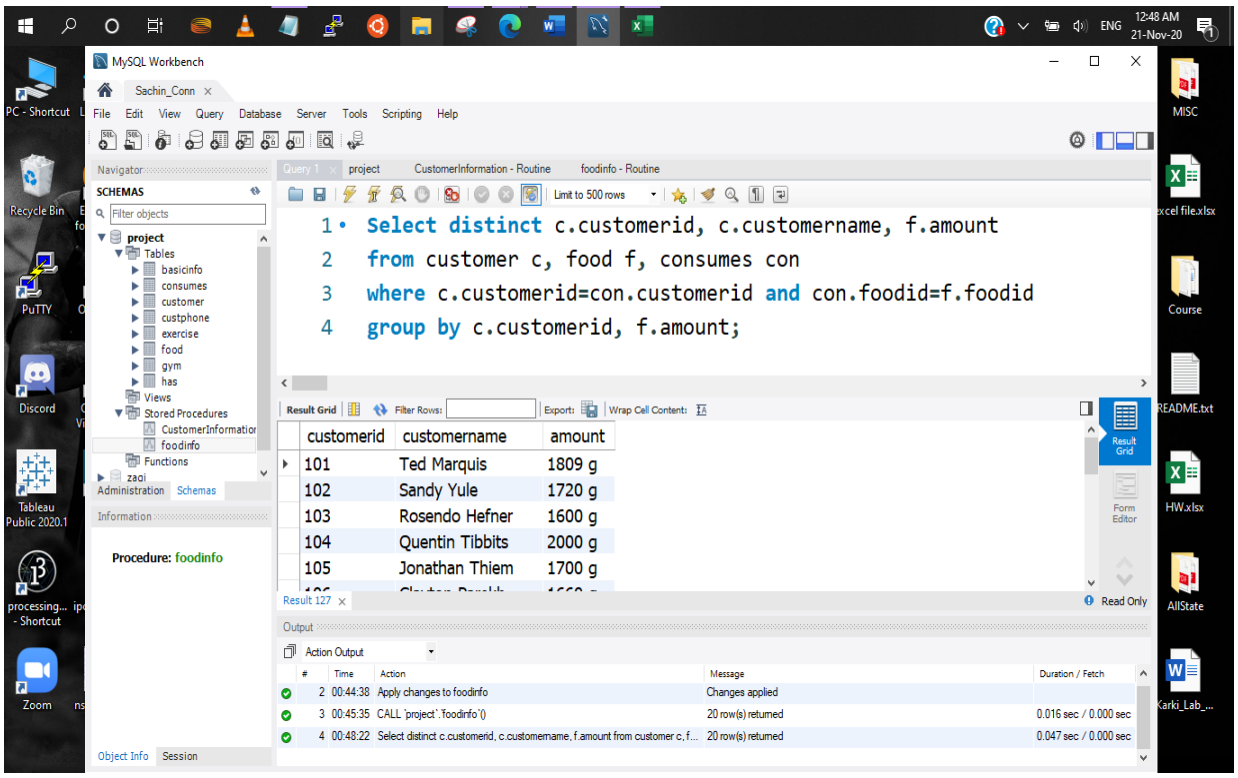


Query 2. Retrieve the basic info of the customer and order by email

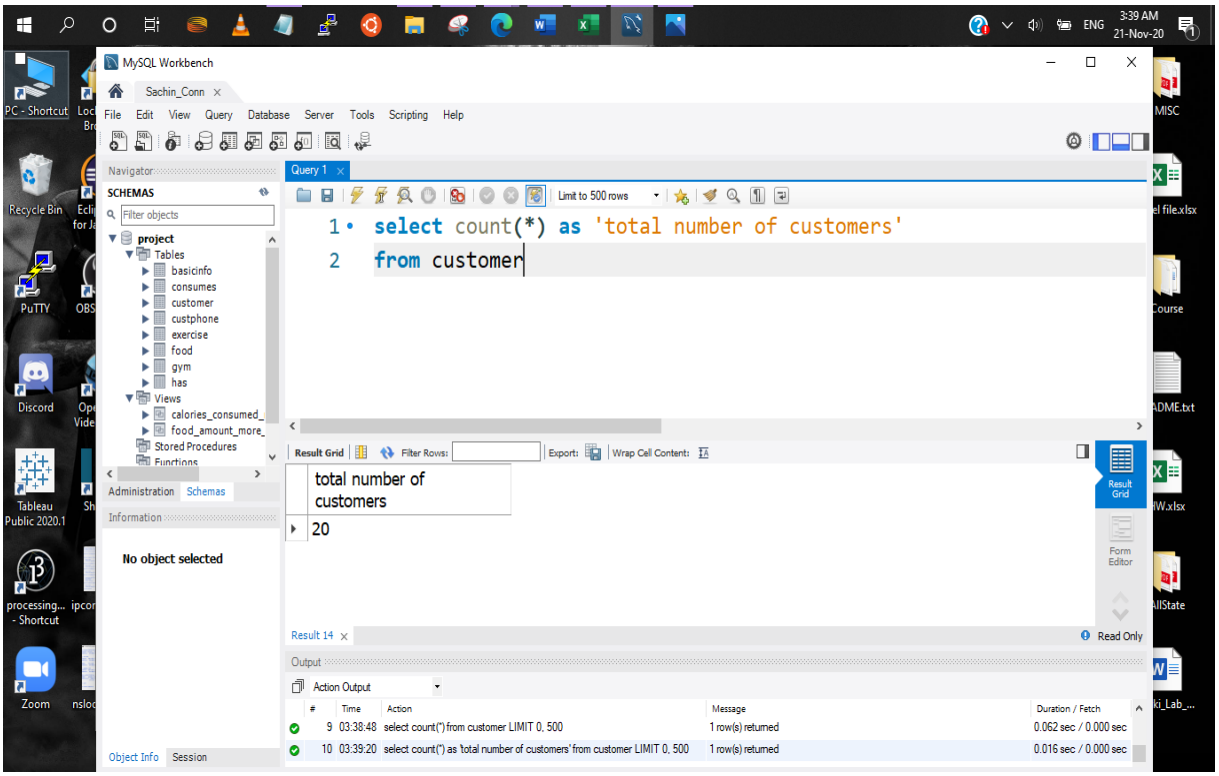


4 medium difficulty queries. Queries that use composite condition for selection, computations, aggregate function and grouping.

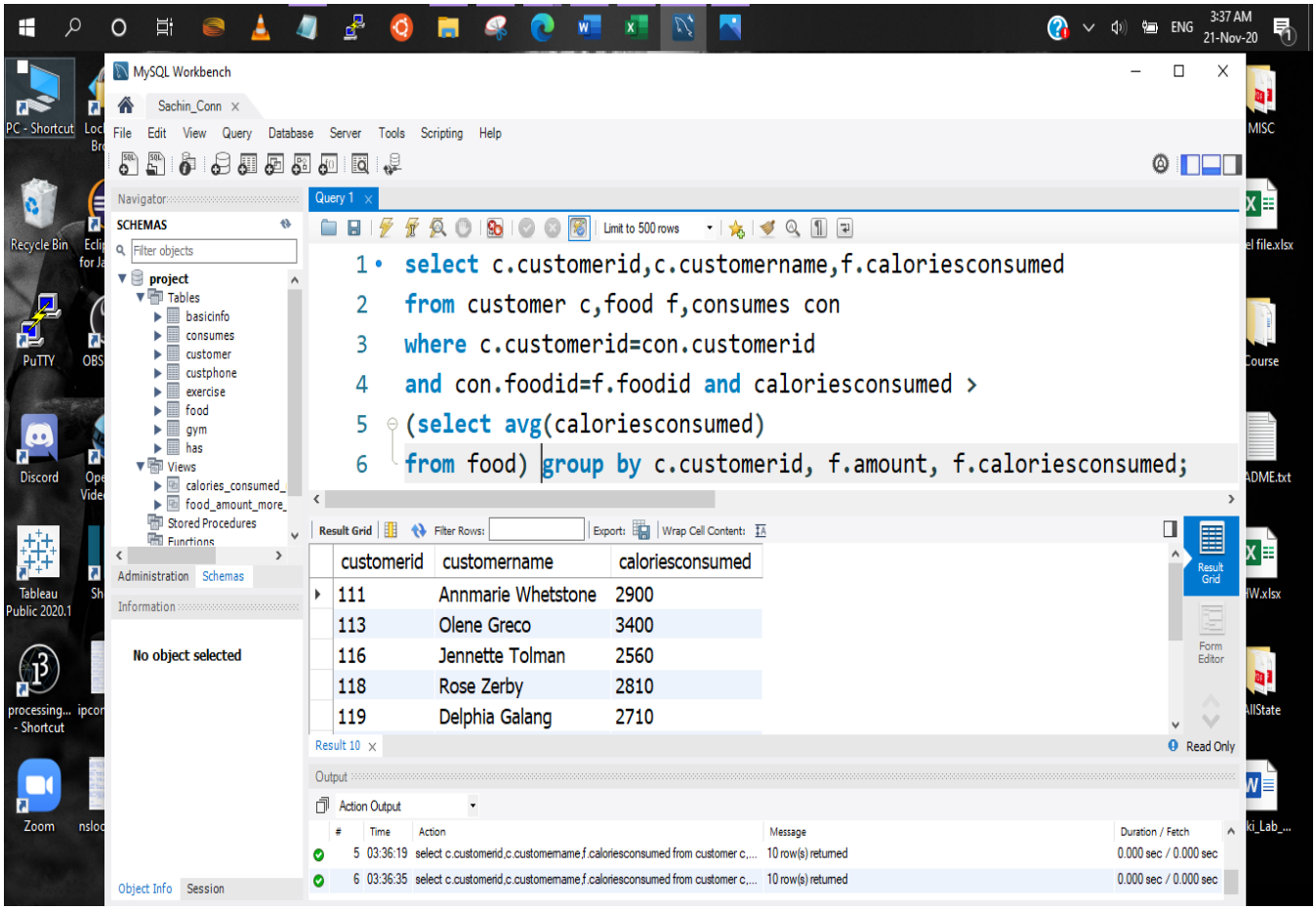
Query 3: For each user retrieve customerid, customername and amount of food while grouping the results by customerid and amount.



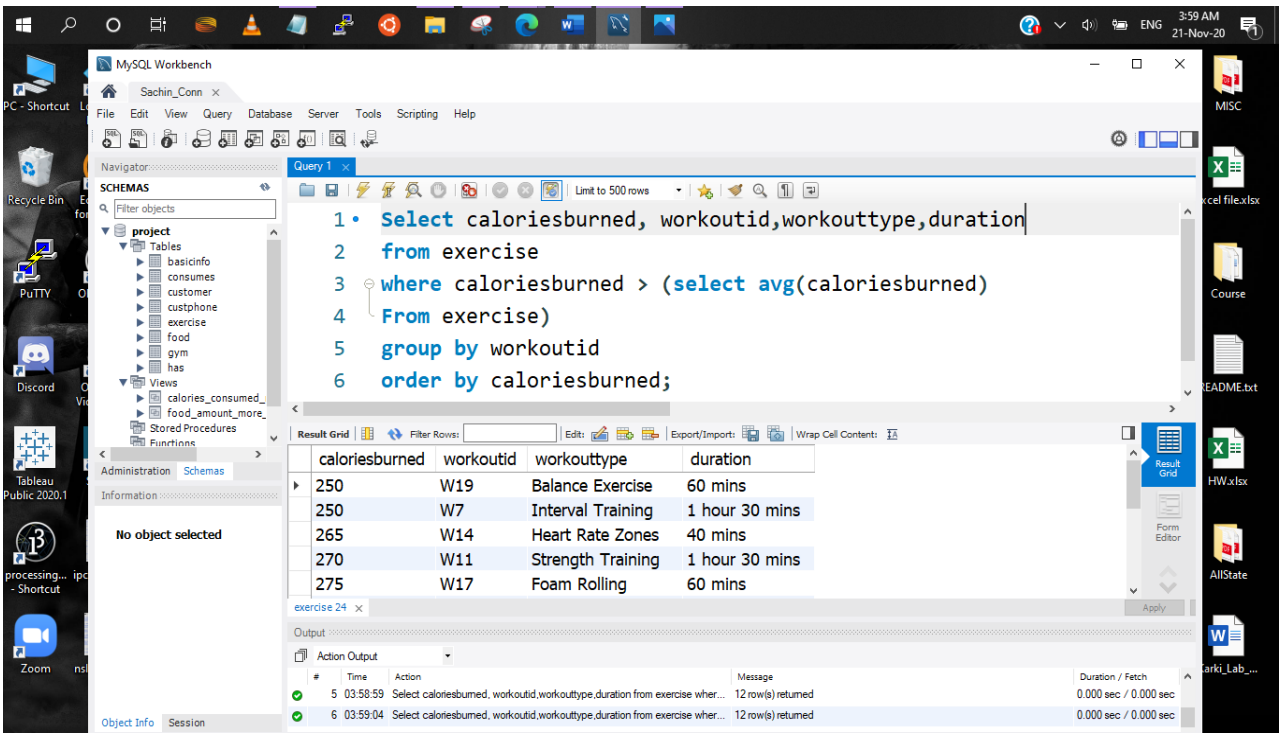
Query 4: Retrieve number of available customers as ‘total number of customers’



Query 5: Retrieve customerid, customername and average caloriesconsumed of each customer. Also, group the table by customerid, amount and caloriesconsumed

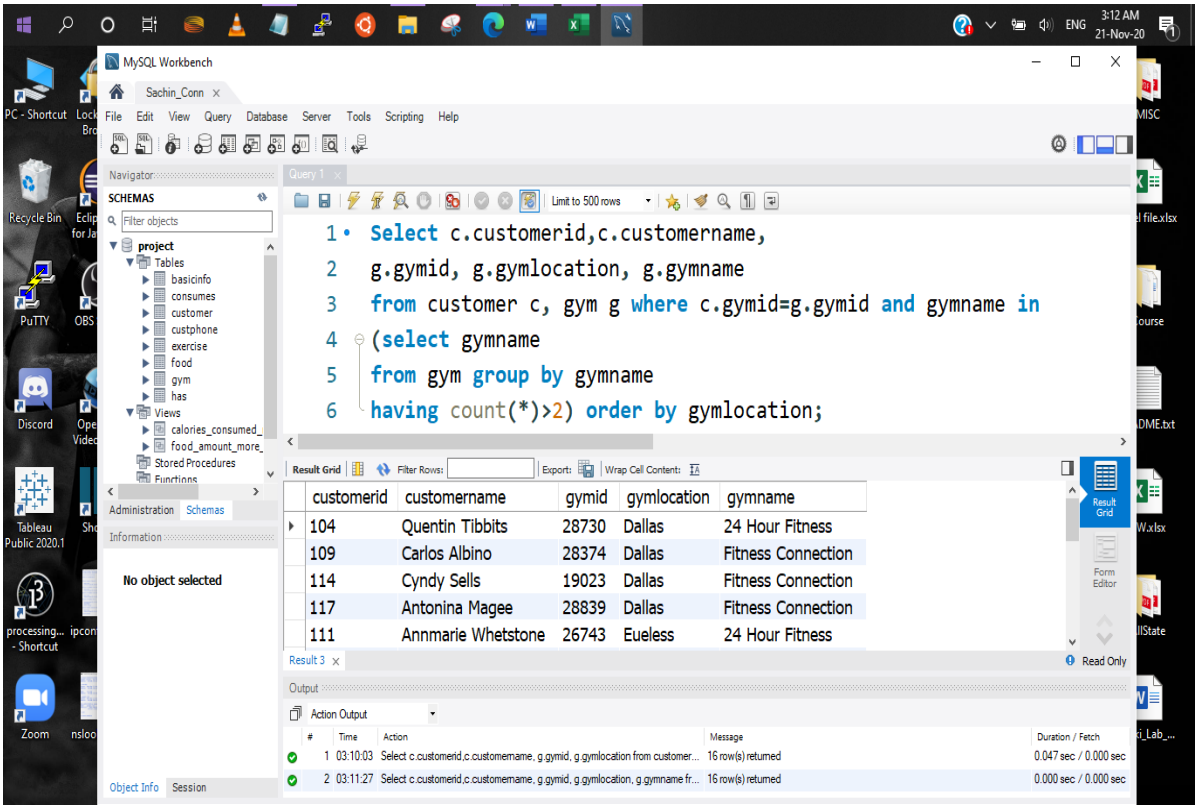


Query 6: Retrieve workoutid, workouttype and duration with caloriesburned on average. Also, group and sort the result by workoutid and caloriesburned respectively.

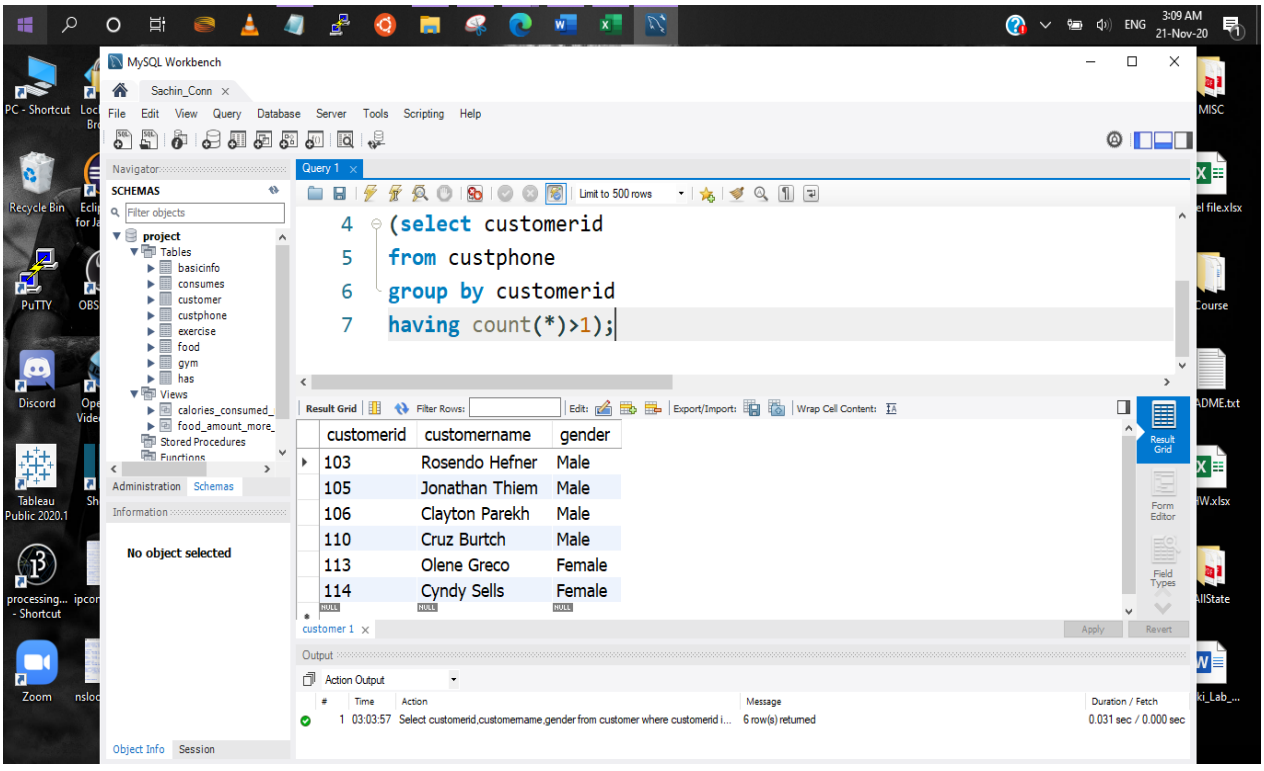


2 query that uses subquery.

Query 7: For each customername, retrieve customerid, gymid, gymlocation where number of gymlocation is more than 2. Also group the results by gymname

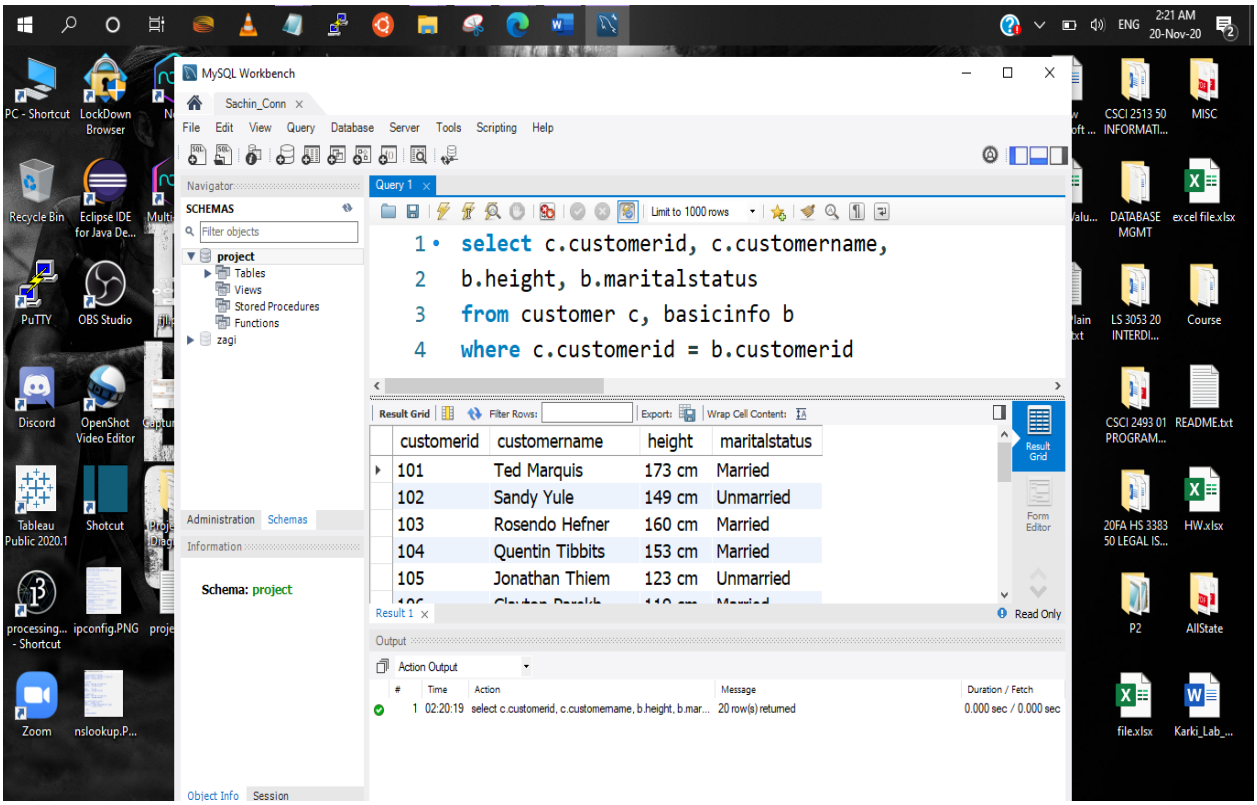


Query 8: Retrieve customerid, customername of users having more than one phone number

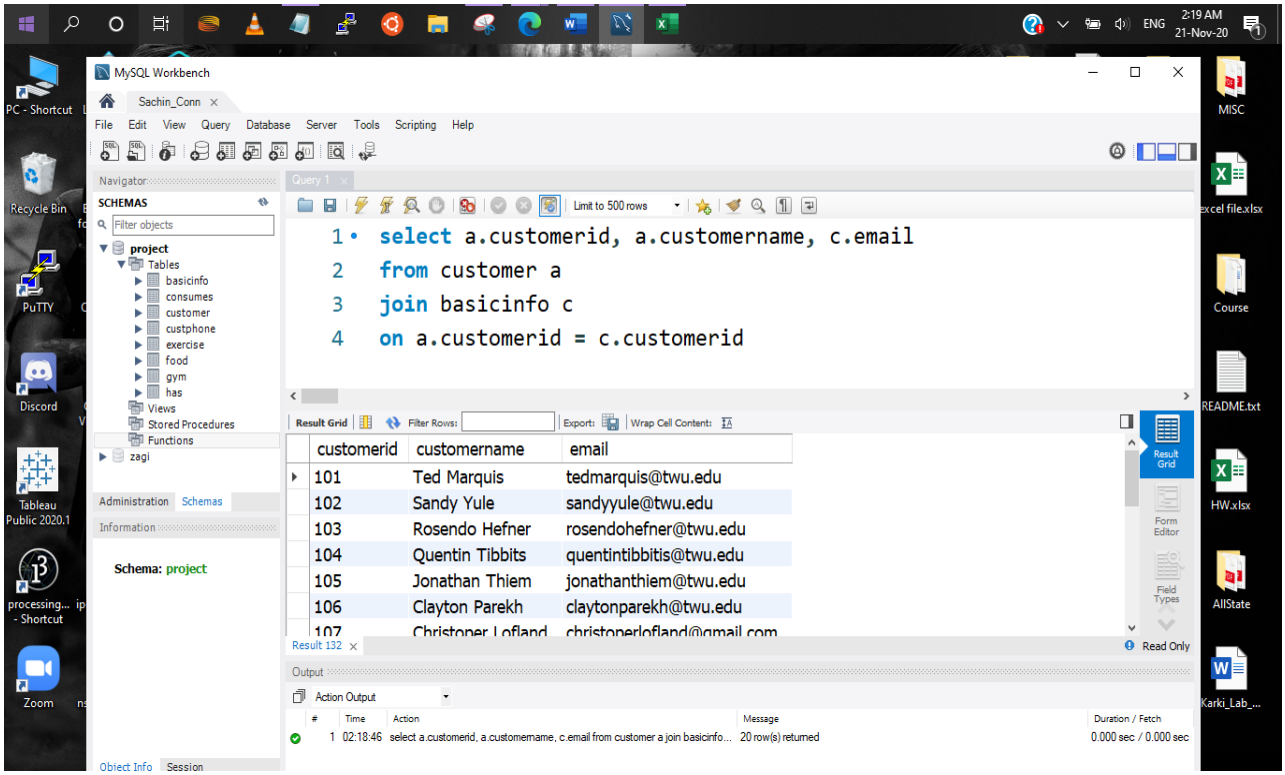


2 queries that uses join (1 inner join, 1 left or right outer join).

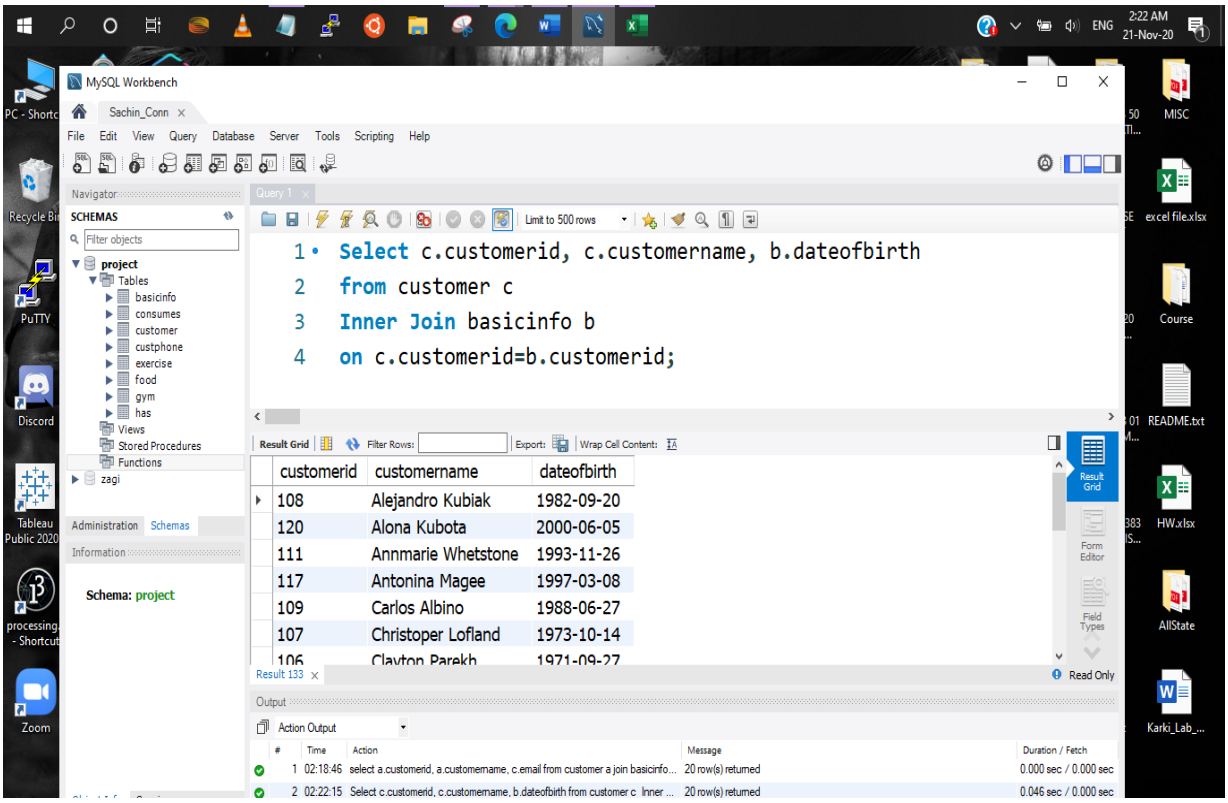
Query 9: Using self-join, retrieve customer id, customer name, height and their marital status



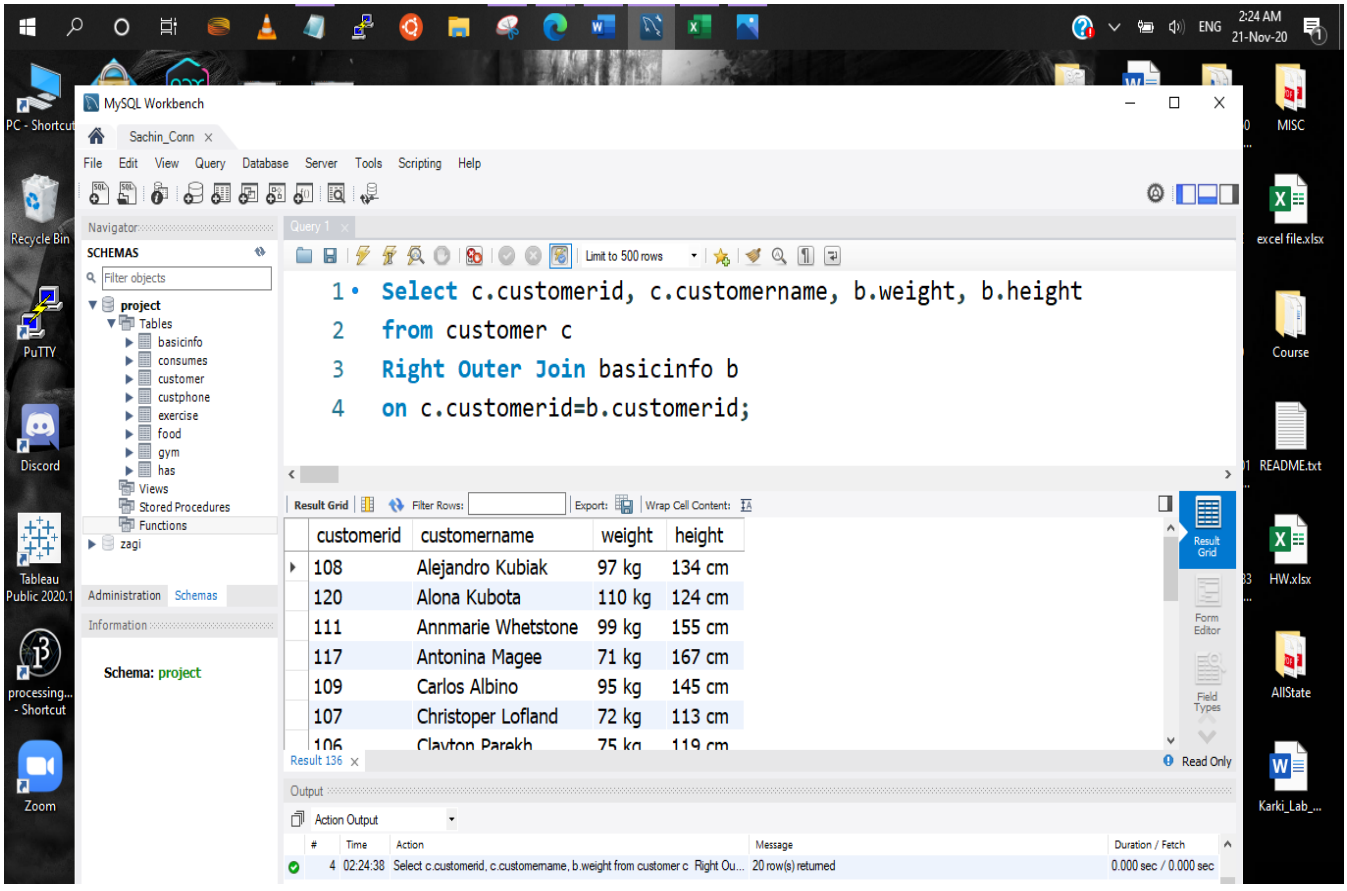
Query 10: Using join, for each customer retrieve customerid, customername and email



Query 11: Using inner join, for each customer retrieve customerid, customername and dateofbirth

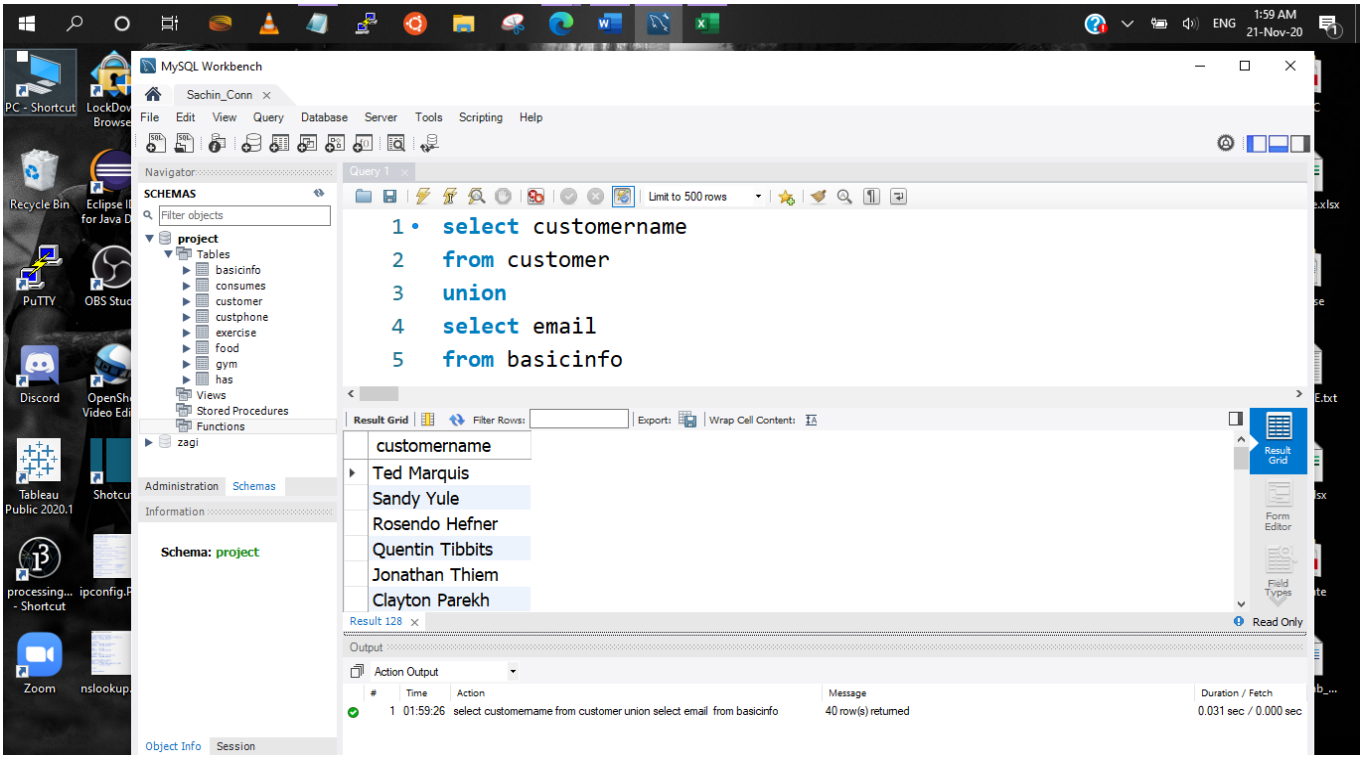


Query 12: Using right outer join, for each customer retrieve customerid, customername, weight and height

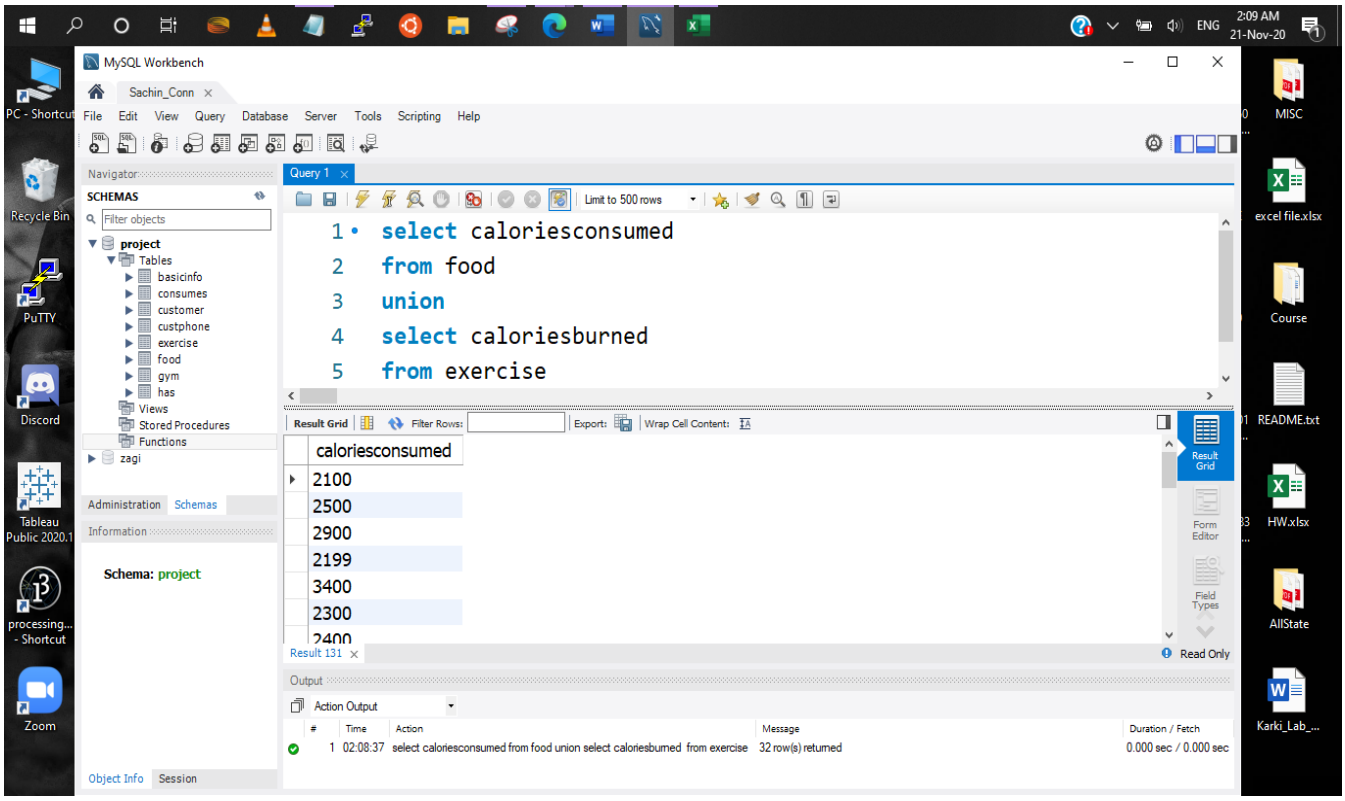


2 query that uses union.

Query 13: For each customer, retrieve customer name and email using union

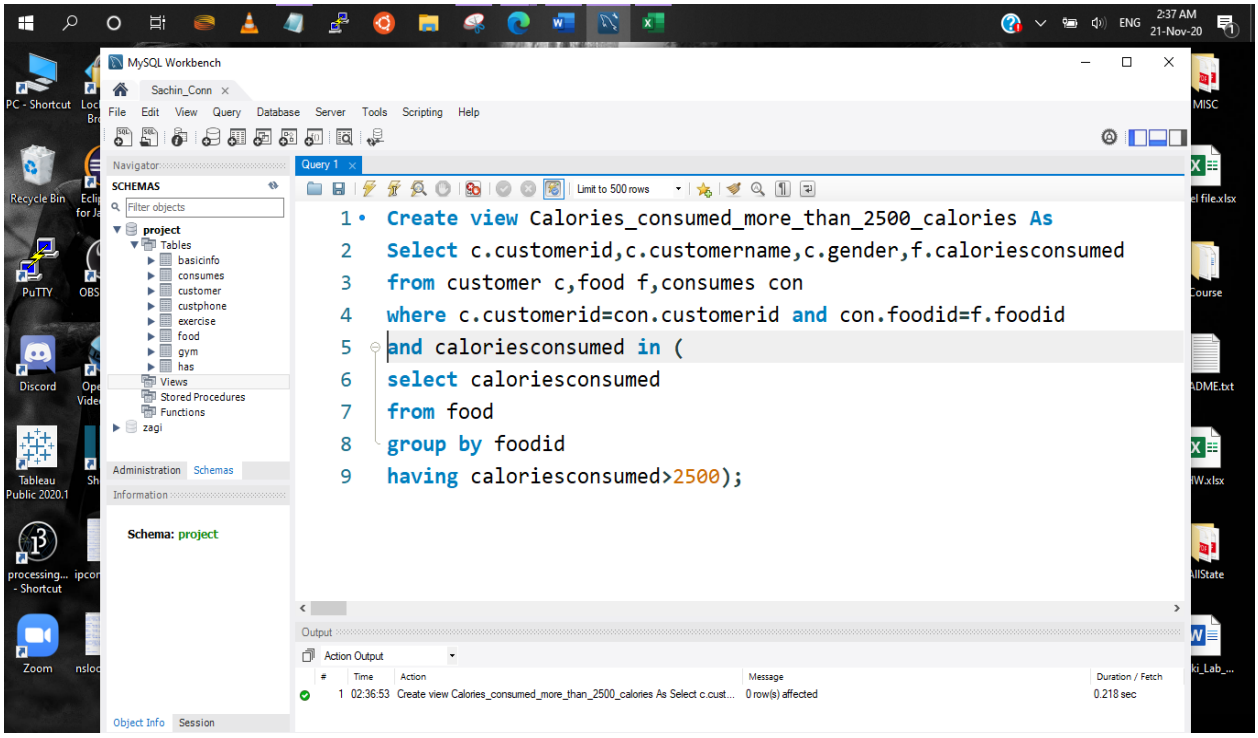


Query 14: Retrieve calories consumed and calories burned using union

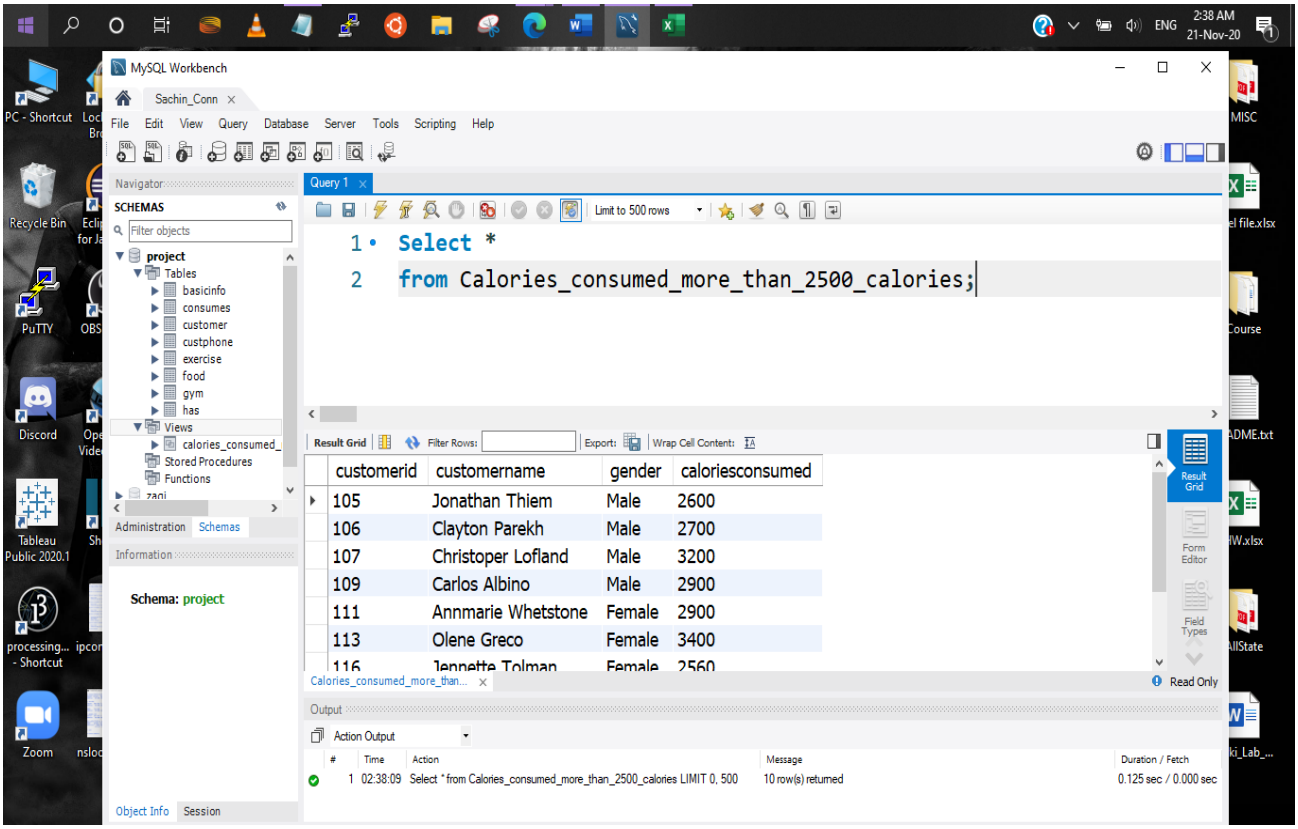


2 view (query must use join – hint: you can use one of the queries from the previous question).

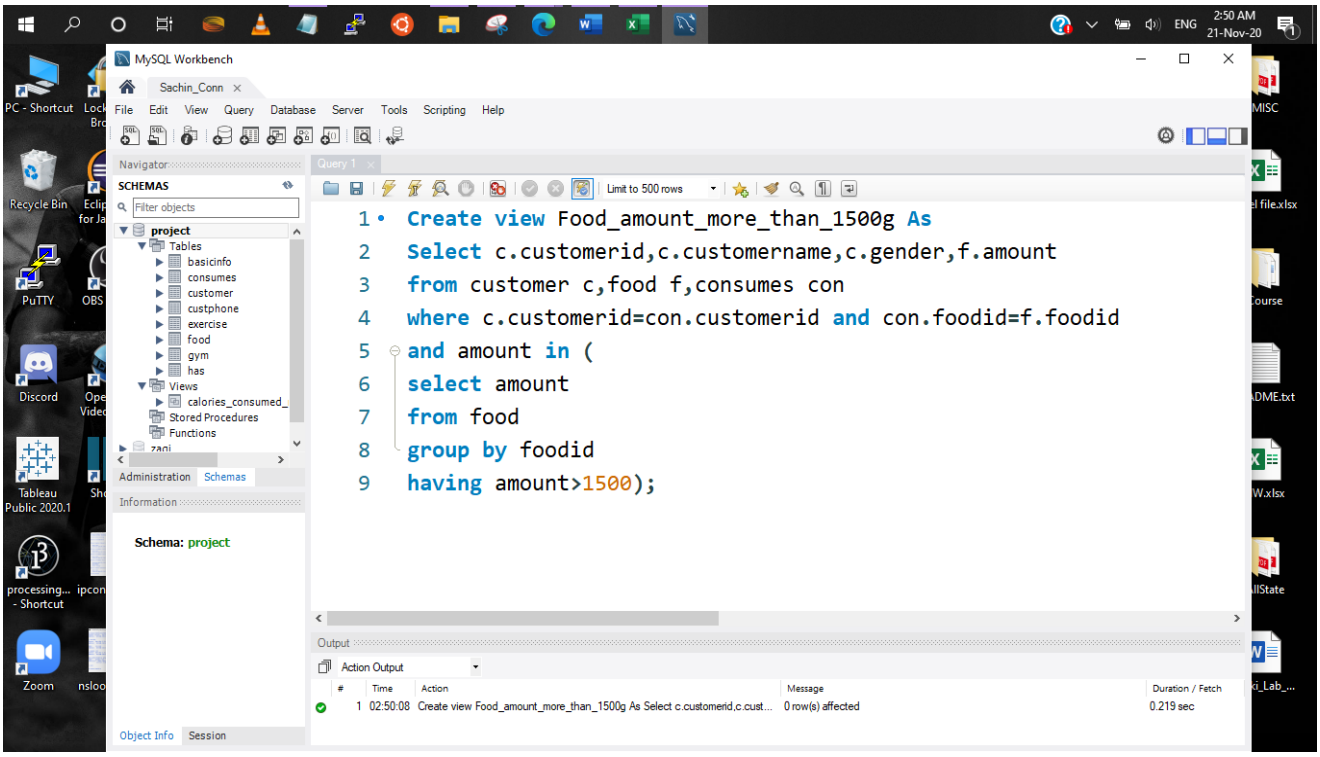
Query 15: using create view, retrieve customerid, customername, gender of customer whose calories consumption is more than 2500.



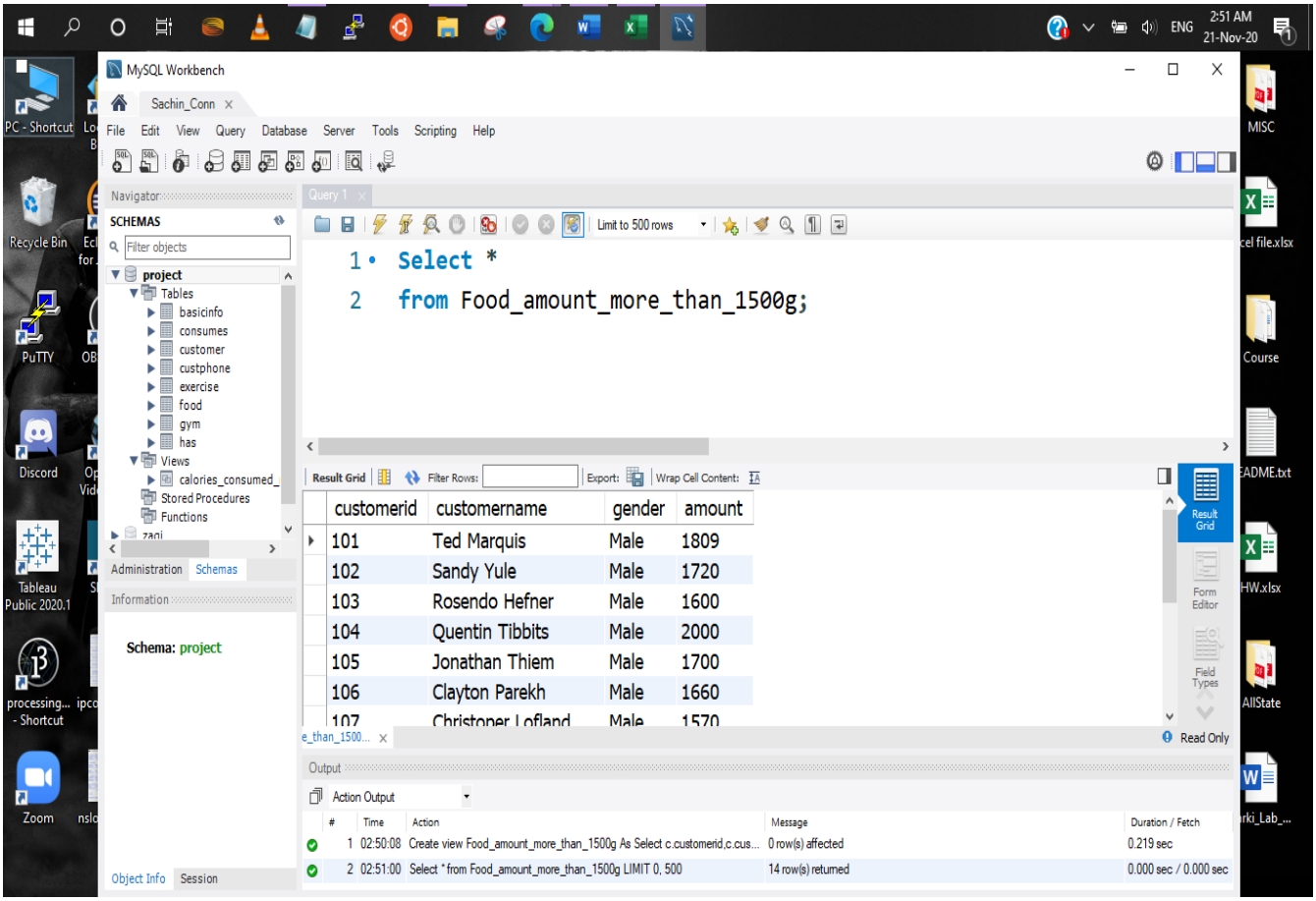
Results:



Query 16: Using create view, retrieve customerid, customername, gender of a customer whose amount of food intake is more than 1500 g.



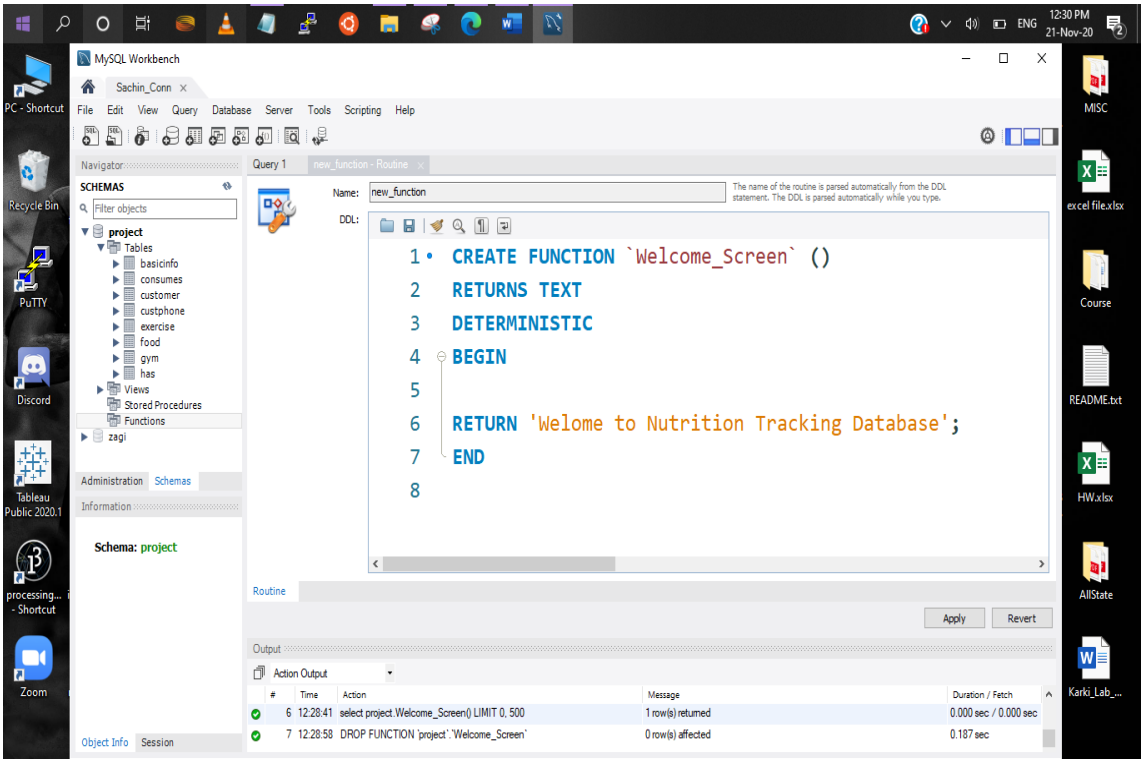
Results:



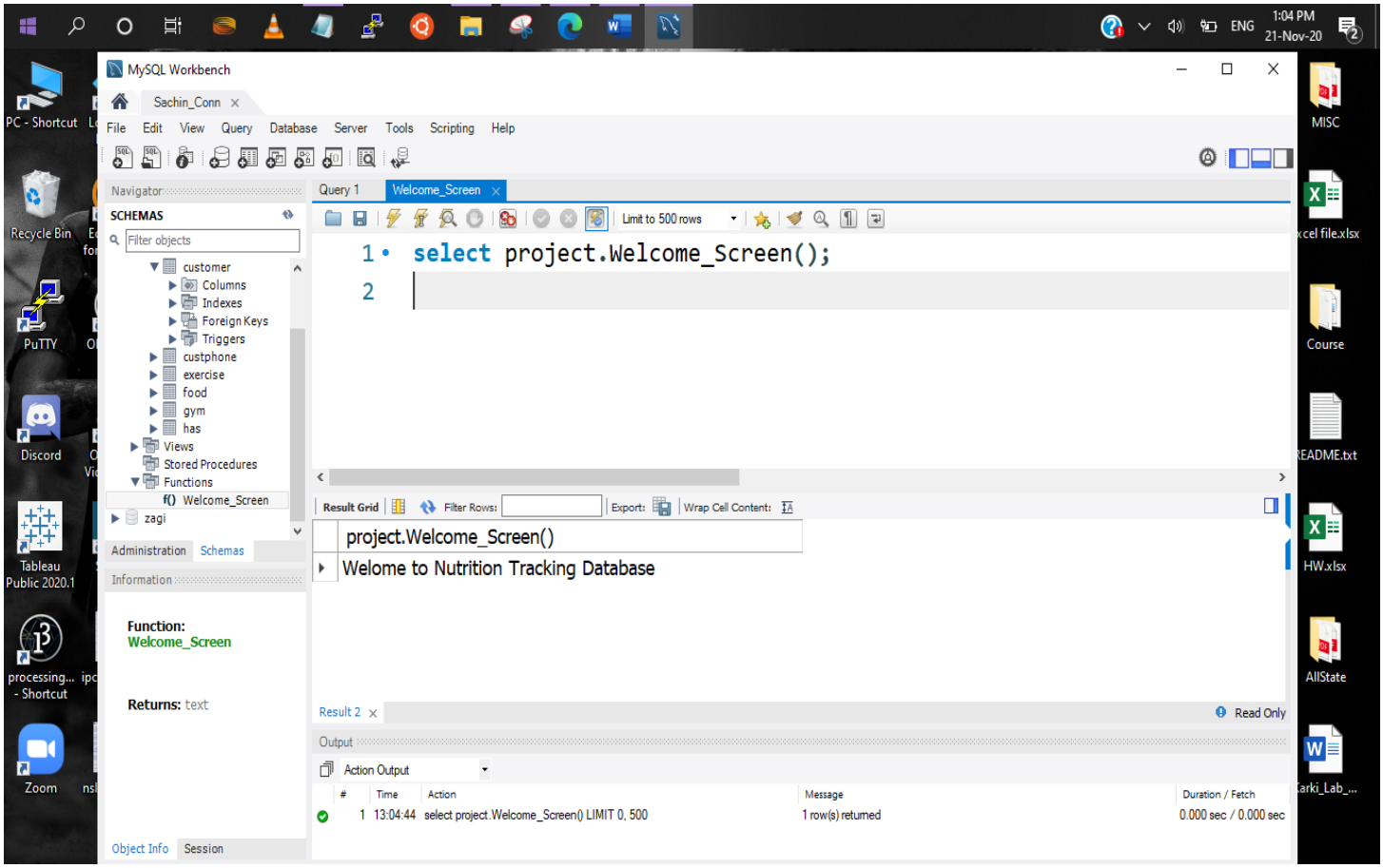
2 custom stored function.

Query 17: Create a custom stored function to display name of the database system as welcome screen

Query Code:

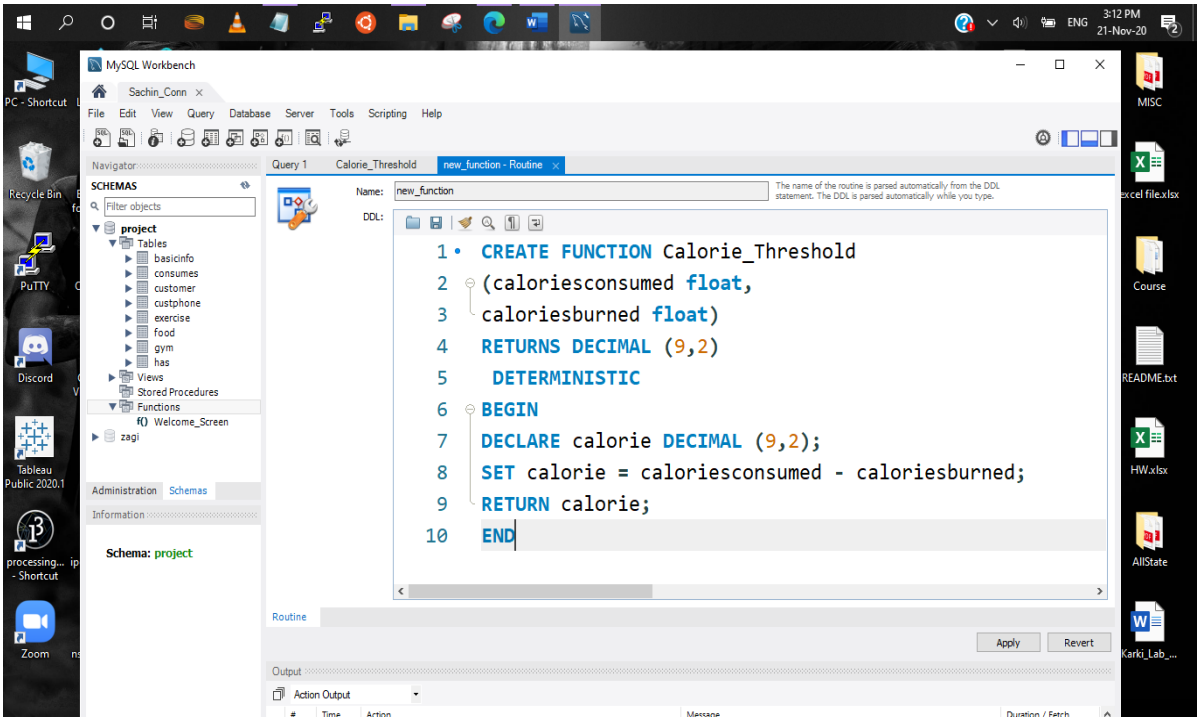


Result:

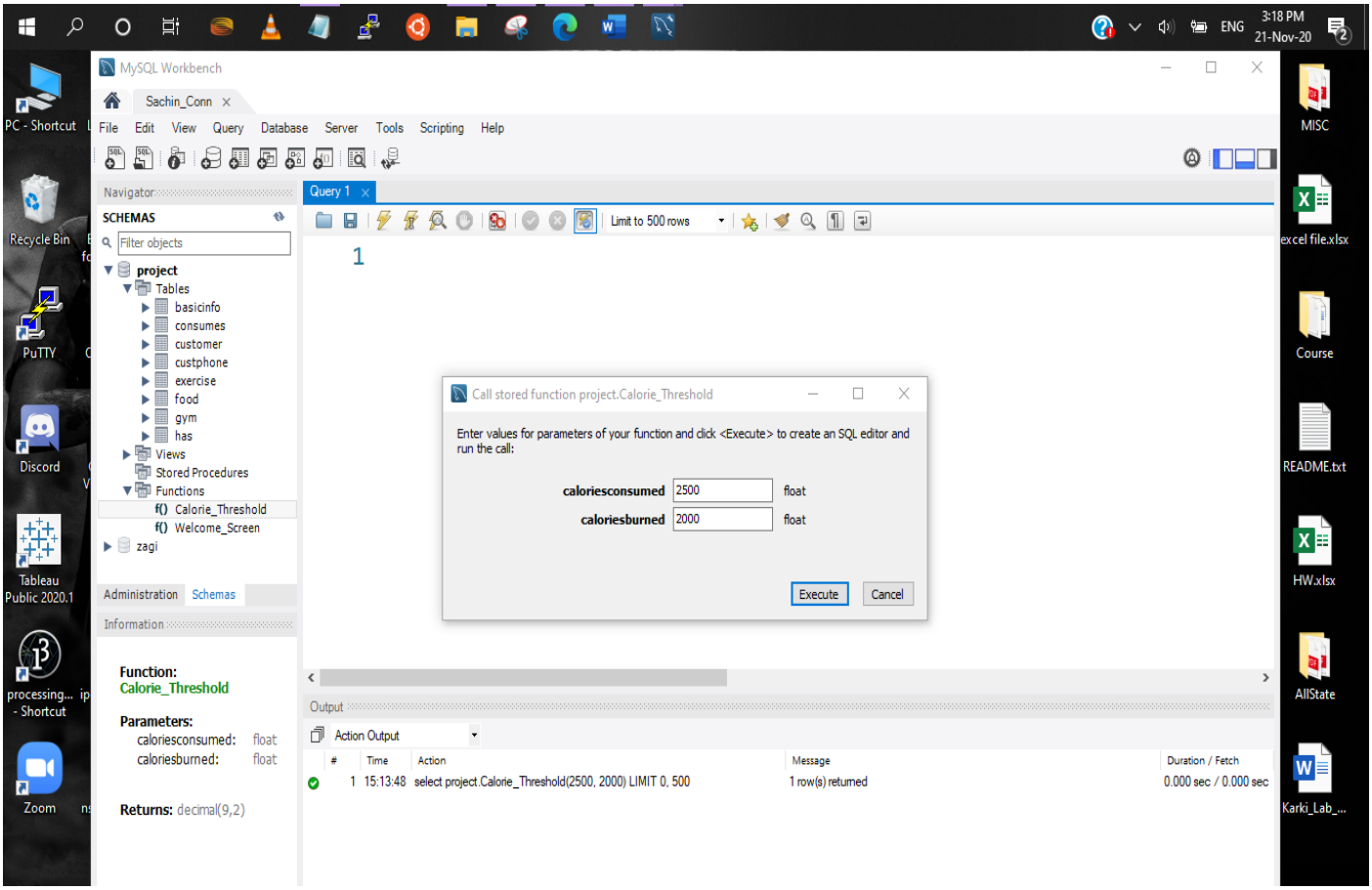


Query 18: Custom stored function to calculate threshold of calorie intake.

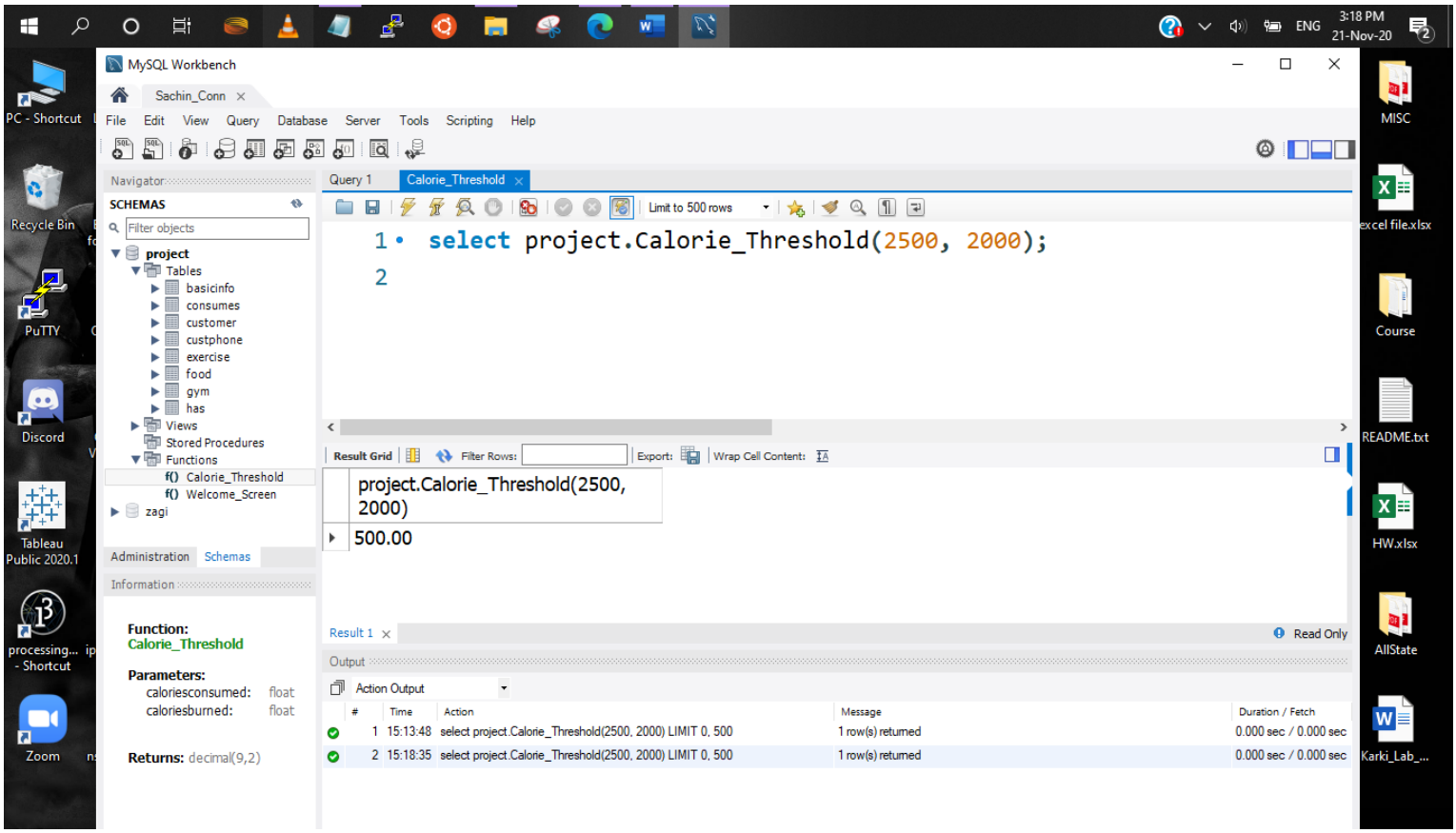
Query Code:



Query Prompt:



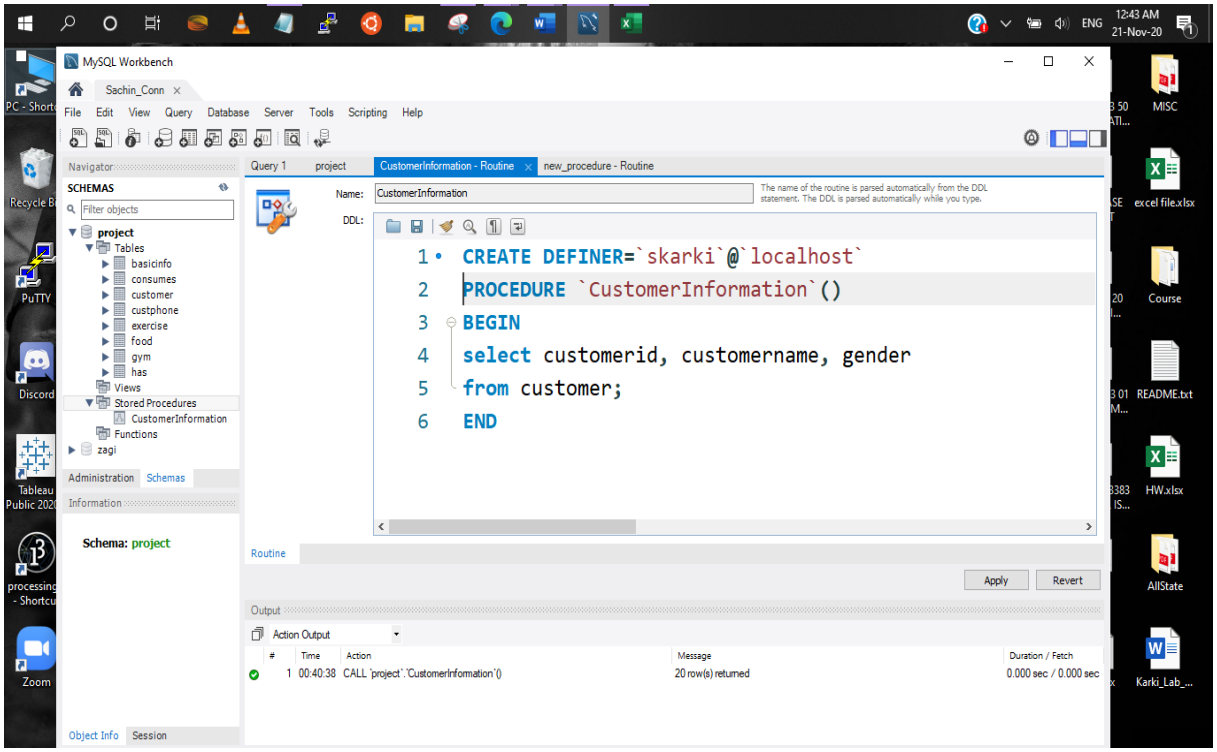
Query Result:



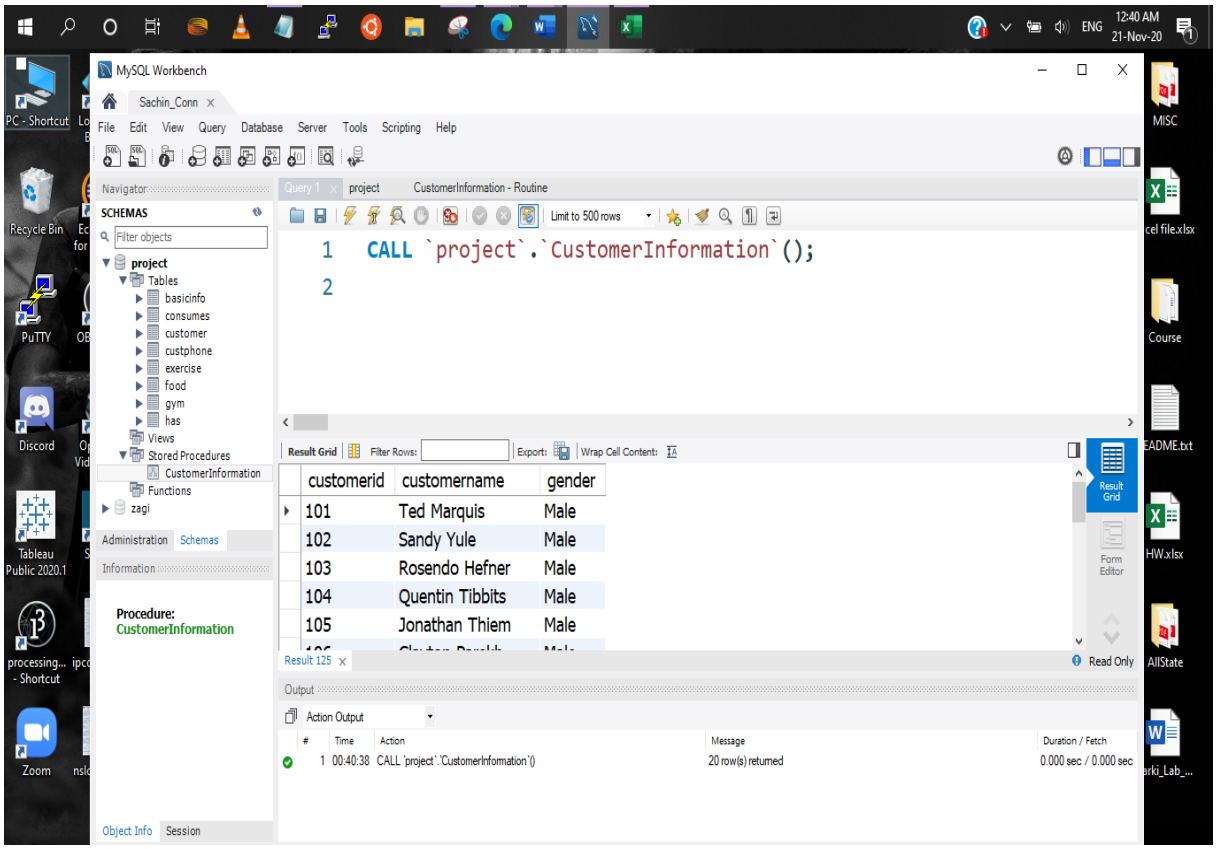
2 custom stored procedure.

Query 19: Custom stored procedure to display entire records of customer information

Query Code:

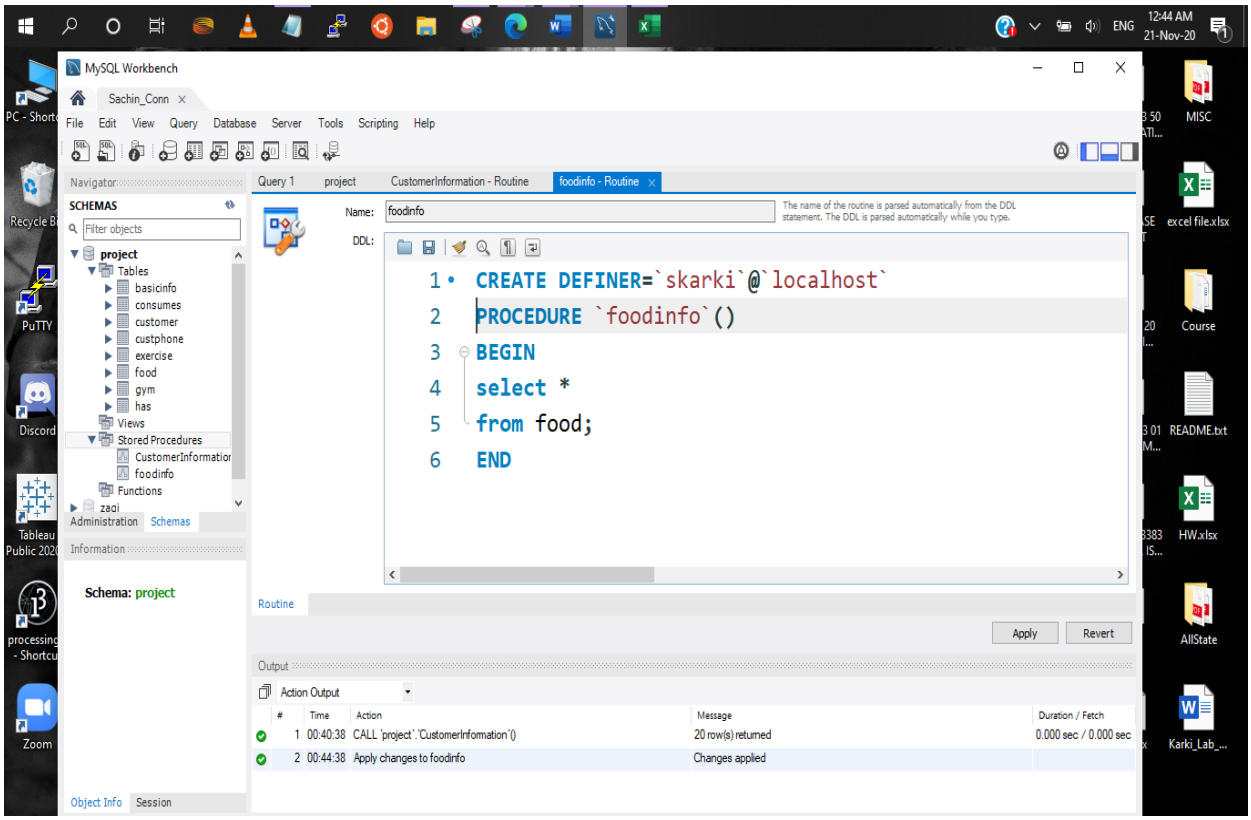


Query Result:

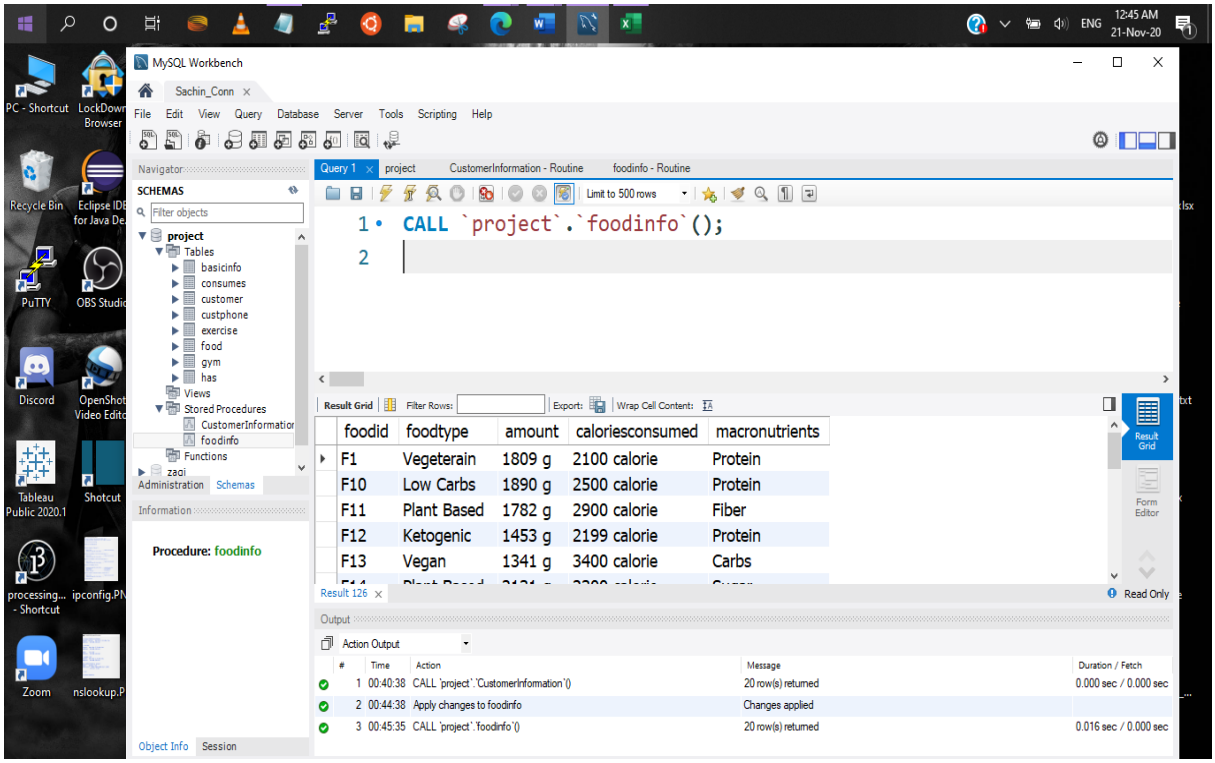


Query 20: Custom stored procedure to display entire contents of food.

Query code:



Query Results:



5. Summary

The proposed database has a potential to become an ideal prototype for a fully functioning database system . As it has presented 20 queries to demonstrate the reliability and accuracy of the database itself there is still room for various interpretations and iterations of same as well as additional queries. To sum up, the Nutrition Tracking Database is flexible and supportive of additional records and information.