Nalongsone Danddank Student ID : 14958950 StarID: jf3893pd

Email: [nalongsone.danddank@my.metrostate.edu](mailto:nalongsone.danddank@my.metrostate.edu)\

**Metropolitan State University**

**ICS 311 — Database Implementation on MySQL**

**Total: 24 Points**

**Part 1 (12 Points):**

In this part of the project you are required to implement your database on MySQL. Your implementation should include the following:

1- Create all the tables that you specified in your relational schema (Part 2, of Step 2). Make sure to include primary keys and foreign keys.

*drop database if exists Restaurant\_Manegement;*

*create database Restaurant\_Management;*

*use Restaurant\_Management;*

*drop table if exists User;*

*create table User(*

*`id` int not null auto\_increment,*

*`fName` varchar(50) not null,*

*`lName` varchar(50) NULL DEFAULT NULL,*

*`phone` int not NULL unique,*

*`email` varchar(50) NULL unique,*

*`pwd` varchar(32) NOT NULL,*

*`user\_role` varchar(32),*

*`line` VARCHAR(50) NULL DEFAULT NULL,*

*`city` VARCHAR(50) NULL DEFAULT NULL,*

*`province` VARCHAR(50) NULL DEFAULT NULL,*

*`country` VARCHAR(50) NULL DEFAULT NULL,*

*`registeredAt` DATETIME NOT NULL,*

*`intro` varchar(300) NULL DEFAULT NULL,*

*`bdate` DATETIME NOT NULL,*

*`gender` VARCHAR(10) NOT NULL,*

*`salary` FLOAT NOT NULL DEFAULT 0,*

*PRIMARY KEY (`id`),*

*UNIQUE INDEX `uq\_mobile` (`phone` ASC),*

*UNIQUE INDEX `uq\_email` (`email` ASC)*

*);*

*drop table if exists Ingredient;*

*create table Ingredient(*

*`id` int not null auto\_increment,*

*`userId` int not null,*

*`title` VARCHAR(75) NOT NULL,*

*`type` VARCHAR(75) NOT NULL ,*

*`quantity` FLOAT NOT NULL DEFAULT 0,*

*`unit` INT NOT NULL DEFAULT 0,*

*`content` TEXT NULL DEFAULT NULL,*

*PRIMARY KEY (`id`),*

*INDEX `idx\_ingredient\_user` (`userId` ASC),*

*CONSTRAINT `fk\_ingredient\_user`*

*FOREIGN KEY (`userId`)*

*REFERENCES User (`id`)*

*ON DELETE RESTRICT*

*ON UPDATE NO ACTION*

*);*

*drop table if exists Item;*

*create table Item(*

*`id` INT NOT NULL AUTO\_INCREMENT,*

*`userId` INT NOT NULL,*

*`title` VARCHAR(75) NOT NULL,*

*`type` VARCHAR(75) NOT NULL ,*

*`price` FLOAT NOT NULL DEFAULT 0,*

*`quantity` FLOAT NOT NULL DEFAULT 0,*

*`unit` INT NOT NULL DEFAULT 0,*

*`recipe` TEXT NULL DEFAULT NULL,*

*`instructions` TEXT NULL DEFAULT NULL,*

*`content` TEXT NULL DEFAULT NULL,*

*PRIMARY KEY (`id`),*

*INDEX `idx\_item\_user` (`userId` ASC),*

*CONSTRAINT `fk\_item\_user`*

*FOREIGN KEY (`userId`)*

*REFERENCES User(`id`)*

*ON DELETE RESTRICT*

*ON UPDATE NO ACTION*

*);*

*drop table if exists Recipe;*

*create table Recipe(*

*`id` INT NOT NULL AUTO\_INCREMENT,*

*`itemId` INT NOT NULL,*

*`ingredientId` INT NOT NULL,*

*`quantity` FLOAT NOT NULL DEFAULT 0,*

*`unit` INT NOT NULL DEFAULT 0,*

*`instructions` TEXT NULL DEFAULT NULL,*

*PRIMARY KEY (`id`),*

*INDEX `idx\_recipe\_item` (`itemId` ASC),*

*UNIQUE INDEX `uq\_recipe\_item\_ingredient` (`itemId` ASC, `ingredientId` ASC),*

*CONSTRAINT `fk\_recipe\_item`*

*FOREIGN KEY (`itemId`)*

*REFERENCES Item(`id`),*

*INDEX `idx\_recipe\_ingredient` (`ingredientId` ASC),*

*CONSTRAINT `fk\_recipe\_ingredient`*

*FOREIGN KEY (`ingredientId`)*

*REFERENCES Ingredient(`id`)*

*ON DELETE RESTRICT*

*ON UPDATE NO ACTION*

*);*

*drop table if exists Menu;*

*create table Menu(*

*`id` INT NOT NULL AUTO\_INCREMENT,*

*`userId` INT NOT NULL,*

*`title` VARCHAR(75) NOT NULL,*

*`type` VARCHAR(75) NOT NULL ,*

*`content` TEXT NULL DEFAULT NULL,*

*PRIMARY KEY (`id`),*

*INDEX `idx\_menu\_user` (`userId` ASC),*

*CONSTRAINT `fk\_menu\_user`*

*FOREIGN KEY (`userId`)*

*REFERENCES User (`id`)*

*ON DELETE RESTRICT*

*ON UPDATE NO ACTION*

*);*

*drop table if exists Menu\_Item;*

*create table Menu\_Item(*

*`id` INT NOT NULL AUTO\_INCREMENT,*

*`menuId` INT NOT NULL,*

*`itemId` INT NOT NULL,*

*`active` TINYINT(1) NOT NULL DEFAULT 1,*

*PRIMARY KEY (`id`),*

*INDEX `idx\_menu\_item\_menu` (`menuId` ASC),*

*INDEX `idx\_menu\_item\_item` (`itemId` ASC),*

*UNIQUE INDEX `uq\_menu\_item` (`menuId` ASC, `itemId` ASC),*

*CONSTRAINT `fk\_menu\_item\_menu`*

*FOREIGN KEY (`menuId`)*

*REFERENCES Menu (`id`),*

*CONSTRAINT `fk\_menu\_item\_item`*

*FOREIGN KEY (`itemId`)*

*REFERENCES Item (`id`)*

*ON DELETE RESTRICT*

*ON UPDATE NO ACTION*

*);*

*drop table if exists Item\_User;*

*create table Item\_User(*

*`id` INT NOT NULL AUTO\_INCREMENT,*

*`itemId` INT NOT NULL,*

*`userId` INT NOT NULL,*

*`active` TINYINT(1) NOT NULL DEFAULT 1,*

*PRIMARY KEY (`id`),*

*INDEX `idx\_item\_user\_item` (`itemId` ASC),*

*INDEX `idx\_item\_user` (`userId` ASC),*

*UNIQUE INDEX `uq\_item\_user` (`itemId` ASC, `userId` ASC),*

*CONSTRAINT `fk\_item\_user\_item`*

*FOREIGN KEY (`itemId`)*

*REFERENCES Item (`id`),*

*CONSTRAINT `fk\_item\_user\_user`*

*FOREIGN KEY (`userId`)*

*REFERENCES User (`id`)*

*ON DELETE CASCADE*

*ON UPDATE NO ACTION*

*);*

*drop table if exists `Order`;*

*create table `Order`(*

*`id` INT NOT NULL AUTO\_INCREMENT,*

*`userId` INT NULL DEFAULT NULL,*

*`status` VARCHAR(30) NOT NULL,*

*`subTotal` FLOAT NOT NULL DEFAULT 0,*

*`itemDiscount` FLOAT NOT NULL DEFAULT 0,*

*`tax` FLOAT NOT NULL DEFAULT 0,*

*`shippingFee` FLOAT NOT NULL DEFAULT 0,*

*`total` FLOAT NOT NULL DEFAULT 0,*

*`discount` FLOAT NOT NULL DEFAULT 0,*

*`grandTotal` FLOAT NOT NULL DEFAULT 0,*

*`createdAt` DATETIME NOT NULL,*

*`updatedAt` DATETIME NULL DEFAULT NULL,*

*`content` TEXT NULL DEFAULT NULL,*

*PRIMARY KEY (`id`),*

*INDEX `idx\_order\_user` (`userId` ASC),*

*CONSTRAINT `fk\_order\_user`*

*FOREIGN KEY (`userId`)*

*REFERENCES User (`id`)*

*ON DELETE NO ACTION*

*ON UPDATE NO ACTION*

*);*

*drop table if exists `Order\_Item`;*

*create table `Order\_Item`(*

*`id` INT NOT NULL AUTO\_INCREMENT,*

*`orderId` INT NOT NULL,*

*`itemId` INT NOT NULL,*

*`price` FLOAT NOT NULL DEFAULT 0,*

*`discount` FLOAT NOT NULL DEFAULT 0,*

*`quantity` FLOAT NOT NULL DEFAULT 0,*

*`unit` INT NOT NULL DEFAULT 0,*

*`createdAt` DATETIME NOT NULL,*

*`updatedAt` DATETIME NULL DEFAULT NULL,*

*`content` TEXT NULL DEFAULT NULL,*

*PRIMARY KEY (`id`),*

*INDEX `idx\_order\_item\_order` (`orderId` ASC),*

*INDEX `idx\_order\_item\_item` (`itemId` ASC),*

*CONSTRAINT `fk\_order\_item\_order`*

*FOREIGN KEY (`orderId`)*

*REFERENCES `Order`(`id`),*

*CONSTRAINT `fk\_order\_item\_item`*

*FOREIGN KEY (`itemId`)*

*REFERENCES `Item` (`id`)*

*ON DELETE NO ACTION*

*ON UPDATE NO ACTION*

*);*

*drop table if exists `Transaction`;*

*create table `Transaction`(*

*`id` INT NOT NULL AUTO\_INCREMENT,*

*`userId` INT NOT NULL,*

*`orderId` INT NOT NULL,*

*`code` VARCHAR(100) NOT NULL,*

*`type` VARCHAR(75) NOT NULL,*

*`mode` VARCHAR(75) NOT NULL,*

*`status` VARCHAR(30) NOT NULL,*

*`createdAt` DATETIME NOT NULL,*

*`updatedAt` DATETIME NULL DEFAULT NULL,*

*`content` TEXT NULL DEFAULT NULL,*

*PRIMARY KEY (`id`),*

*INDEX `idx\_transaction\_user` (`userId` ASC),*

*INDEX `idx\_transaction\_order` (`orderId` ASC),*

*CONSTRAINT `fk\_transaction\_user`*

*FOREIGN KEY (`userId`)*

*REFERENCES `User`(`id`),*

*CONSTRAINT `fk\_transaction\_order`*

*FOREIGN KEY (`orderId`)*

*REFERENCES `Order` (`id`)*

*ON DELETE NO ACTION*

*ON UPDATE NO ACTION*

*);*

1. Populate all tables with data of your choice. Make sure that each table includes at least 10 rows.

*insert into `User` values*

*(1000, 'Jimmy', 'Smith',16129785228, 'jimmy.smith@email.usa', '123456','admin','1087 Maneland Ave W',*

*'St. Paul', 'MN', 'USA', '2009-08-13', 'I am the boss here.', '1985-06-14', 'male', 999999),*

*(1001, 'Stephen', 'King',18241599931, 'stephen.king@email.usa','123456','chef','8468 North Hilldale Drive',*

*'Fayetteville', 'NC', 'USA', '2011-07-10', 'I am the Chef here.', '1985-05-12', 'male', 7000),*

*(1002, 'David', 'Kerry',15422362865, 'david.kerry@email.usa','123456','chef','1356 Century Ave E',*

*'St, Paul', 'MN', 'USA', '2009-08-13', 'I am the Chef here.', '1989-04-22', 'female', 6500),*

*(1003, 'John', 'Kelvin',19622389383, 'john.kelvin@email.usa','123456','chef','577 Glenwood Dr.',*

*'Palos Verdes Peninsula', 'CA', 'USA', '2009-08-13', 'I am the Chef here.', '1985-05-12', 'male', 4000),*

*(1004, 'Melanie', 'James',16298350615, 'melanie.james@email.usa','123456','chef','573 Blackburn Street',*

*'Osseo', 'MN', 'USA', '2009-08-13', 'I am the Chef here.', '1985-05-12', 'male', 4000),*

*(1005, 'Elizabeth', 'Young',19696180709, 'Elizabeth.Young@email.usa','123456','employee', '936 South Goldfield Court',*

*'Schenectady', 'NY', 'USA','2010-08-13', 'I am employee here.', '1989-08-16', 'female', 3500),*

*(1006, 'Anna', 'Wright',19961614174, 'Anna.Wright@email.usa','123456','employee', '488 S. Golf Rd.',*

*'Westerville', 'AK', 'USA','2015-08-13', 'I am employee here.', '1989-07-16', 'female', 3500),*

*(1007, 'Faith', 'Alsop',15894979029, 'Faith.Alsop@email.usa','123456','employee', '7244 Lees Creek Dr.',*

*'Anchorage', 'OH', 'USA','2015-08-13', 'I am employee here.', '1989-07-16', 'female', 3500),*

*(1008, 'Bella', 'Gibson',18230750098, 'Bella.Gibson@email.usa','123456','employee', '124 Proctor St.',*

*'Revere', 'MA', 'USA','2015-08-13', 'I am employee here.', '1989-07-16', 'female', 3500),*

*(1009, 'Matt', 'Peters',12173802481, 'Matt.Peters@email.usa','123456','employee', '554 Olive St.',*

*'Southaven', 'MS', 'USA','2015-08-13', 'I am employee here.', '1989-07-16', 'female', 3500),*

*(1010, 'Andrea', 'Hart',15872167729, 'Andrea.Hart@email.usa','123456','employee', '9024 Cactus Dr.',*

*'Boca Raton', 'FL', 'USA','2015-08-13', 'I am employee here.', '1989-07-16', 'female', 3500),*

*(1011, 'Melanie', 'Lawrence',14408696376, 'Melanie.Lawrence@email.usa','123456','employee', '89 Whitemarsh Dr.',*

*'Lake Mary', 'FL', 'USA','2015-08-13', 'I am employee here.', '1989-07-16', 'female', 3500),*

*(1012, 'Max', 'Davidson',19226419180, 'Max.Davidson@email.usa','123456','employee', '9827 West Indian Spring Street',*

*'New Bedford', 'MA', 'USA','2015-08-13', 'I am employee here.', '1989-07-16', 'female', 3500),*

*(1013, 'Brandon', 'Mackenzie',12399739247, 'Brandon.Mackenzie@email.usa','123456','employee', '9020 Beach St.',*

*'Norcross', 'GA', 'USA','2015-08-13', 'I am employee here.', '1989-07-16', 'female', 3500);*

*insert into `Ingredient`(userId, title, `type`, quantity) values*

*-- (1010, 'egg', 'available', 100),*

*(1010, 'flour', 'meat', 100),(1010, 'chicken', 'meat', 100),*

*(1010, 'salt', 'available', 100), (1010, 'sugar', 'available', 1000),*

*(1010,'chocolate', 'available', 50), (1010,'vanilla extract', 'available', 10),*

*(1010,'crab', 'meat', 10), (1010,'pork', 'meat', 10), (1010, 'beef', 'meat', 50);*

*insert into `item`(userId, title, `type`,price, quantity ) values*

*(1001, 'PizzaRev’s Cauliflower Power Pizza' ,'Pizza',7.00, 100),*

*(1001, 'Blaze Pizza’s Salad Pizza' ,'Pizza',9.5, 100),*

*(1001, 'Brenz Pizza’s Tres Amigos' ,'Pizza',8, 100),*

*(1001, 'East Side Pie’s Working Man Pizza' ,'Pizza',6, 100),*

*(1002, 'Larkburger’s Chimichurri Bacon Cheeseburger' ,'Burgers',10, 100),*

*(1002, 'Mooyah’s Double Diablo' ,'Burgers',10.75, 100),*

*(1002, 'Grindhouse Killer Burger’s Hillbilly-Style Burger' ,'Burgers',9.55, 100),*

*(1002, 'Gott’s Roadside’s Kimchi Burger' ,'Burgers',8.75, 100),*

*(1002, 'Zombie Burger’s The Walking Ched' ,'Burgers',8.45, 100),*

*(1003, 'Melt Shop’s Fried Chicken Sandwich' ,'Sandwiches',6.65, 200),*

*(1003, 'Which Wich’s Classic Superfood Wich' ,'Sandwiches',7.65, 200),*

*(1003, 'Schlotzsky’s Tex-Mex Brisket Sandwich' ,'Sandwiches',7.95, 100),*

*(1003, 'Wichcraft’s Pork & Slaw' ,'Sandwiches',8.65, 100),*

*(1004, 'Crushed Red’s This Pear Is On Fire Salad' ,'Salads',4.65, 100),*

*(1004, 'MAD Greens’ Genghis Khan Salad' ,'Salads',5.65, 100),*

*(1005, 'Milk','Drink',3.05, 100),(1005, 'Coke','Drink',3.05, 100),*

*(1005, 'Sprite','Drink',2.05, 100),(1005, 'Tea','Drink',4.05, 100),*

*(1005, 'Orange','Drink',2.05, 100),(1005, 'Hot Tea','Drink',3.05, 100),*

*(1005, 'Apple Juice','Drink',4.05, 100),(1005, 'Mango Juice','Drink',5.05, 100),*

*(1006, 'Glass wine red', 'Alcohol',15, 30),(1006, 'Bottle wine white','Alcohol',5.05, 30),*

*(1006, 'Bottle wine red', 'Alcohol',15, 30),(1005, 'Tiramisu','Dessert',6.05, 30),*

*(1006, 'Ice Cream','Dessert',5.15, 30), (1006, 'Chocolate Ice Cream','Dessert',5.55, 30),*

*(1006, 'Chip Cookies Cream','Dessert',4.05, 30);*

*insert into `item\_user`(itemId, userId, `active`) values*

*( 1,1001,'available'),(2,1001,'available'),(3,1001,'available'),(4,1001,'unavailable'),*

*( 5,1002,'available'),(6,1002,'available'),(7,1002,'available'),(8,1002,'unavailable'),(9,1002,'unavailable'),*

*( 10,1003,'available'),( 11,1002,'available');*

*insert into menu(userId, title, `type`) values*

*(1000, 'Pizza','Main'),(1000, 'Burgers','Main'),*

*(1000, 'Sandwiches','Main'),(1000, 'Salads ','Main'),(1000, 'Drink ','Drink'),*

*(1000, 'Alcohol ','Alcohol'),(1000, 'Dessert ','Dessert'),*

*(1010, 'Other', 'Other'),(1010, 'Other', 'Other'),(1010, 'Other', 'Other');*

*insert into menu\_item(menuId, itemId, `active`) values(1, 1, 'available'),(1, 2, 'available'),*

*(1, 3, 'available'),(1, 4, 'available'),(2, 5, 'available'),(2, 6, 'available'),(2, 7, 'available'),*

*(2, 8, 'available'),(2, 9, 'available'),(3, 10, 'available'),(3, 11, 'available'),(3, 12, 'available'),*

*(3, 13, 'available'),(4, 14, 'available'),(4, 15, 'available'),(5, 16, 'available'), (5, 17, 'available'),*

*(5, 18, 'available'),(5, 19, 'available'),(5, 20, 'available'),(5, 21, 'available'),(5, 22, 'available'),*

*(5, 23, 'available'),(6, 24, 'available'),(6, 25, 'available'),(6, 26, 'available'),(7, 27, 'available'),*

*(7, 28, 'available'),(7, 29, 'available'),(7, 30, 'available');*

*insert into `recipe`(itemId, ingredientId) values(1,1),(1,2),(1,3),(2,1),(2,2),(3,1),(3,5),(4,4),(6,5),*

*(8,3),(6,3),(5,6),(3,6),(8,1),(2,9);*

*insert into `order`(id,userId,`status`,tax,shippingFee,createdAt) values*

*(108, 1011, 'New',0.07,8 ,curdate()),(109, 1008, 'New',0.07,8 ,curdate()),(110, 1008, 'New',0.07,8 ,curdate()),*

*(100, 1012, 'New',0.07,8 ,'2020-11-03'),*

*(101, 1007, 'New',0.07,8 ,'2020-11-03'),(102, 1008, 'New',0.07,8 ,'2020-11-03'),*

*(103, 1009, 'New',0.07,8 ,'2020-11-03'),(104, 1010, 'New',0.07,8 ,'2020-11-03'),*

*(105, 1011, 'New',0.07,8 ,'2020-11-03'),(106, 1012, 'New',0.07,8 ,'2020-11-03'),*

*(107, 1007, 'New',0.07,8 ,'2020-11-03');*

*-- -- Create Trigger for updating order table by each time insert new order\_item table*

*-- -- for automatic calculating total payment for customer.*

*drop trigger if exists trg\_order\_order\_item;*

*delimiter $$*

*create trigger trg\_order\_order\_item*

*after insert on order\_item*

*for each row*

*begin*

*declare A float;*

*declare B float;*

*set B = (select subTotal from `order` where id = new.orderId);*

*set A = B + (select price from item where id = new.itemId) \* new.quantity;*

*update `order`*

*set `status` = 'Checkout',*

*subTotal = A,*

*total = A + (A \* tax) + shippingFee,*

*grandTotal = total + total \* new.discount,*

*updatedAt = curdate() where id = new.orderId;*

*end $$*

*delimiter ;*

*-- -- Create Trigger for updating order table by each time when delete order\_item table*

*-- -- for automatic calculating total payment for customer.*

*drop trigger if exists trg\_order\_order\_item\_delete;*

*delimiter $$*

*create trigger trg\_order\_order\_item\_delete*

*before delete on order\_item for each row*

*begin*

*declare A float;*

*declare B float;*

*set B = (select subTotal from `order` where id = old.orderId);*

*set A = B - (select price from item where id = old.itemId) \* old.quantity;*

*update `order`*

*set `status` = 'Checkout',*

*subTotal = A,*

*total = A + (A \* tax) + shippingFee,*

*grandTotal = total + total \* old.discount,*

*updatedAt = curdate() where id = old.orderId;*

*end $$*

*delimiter ;*

*-- -- Create Trigger for updating order table by each time when updated order\_item table*

*-- -- for automatic calculating total payment for customer.*

*drop trigger if exists trg\_order\_order\_item\_update;*

*delimiter $$*

*create trigger trg\_order\_order\_item\_update*

*after update on order\_item for each row*

*begin*

*declare A float;*

*declare B float;*

*declare A\_ float;*

*declare B\_ float;*

*set B = (select subTotal from `order` where id = old.orderId);*

*set A\_ = B - (select price from item where id = old.itemId) \* old.quantity;*

*set A = A\_ + (select price from item where id = new.itemId) \* new.quantity;*

*update `order`*

*set `status` = 'Checkout',*

*subTotal = A,*

*total = A + (A \* tax) + shippingFee,*

*grandTotal = total + total \* old.discount,*

*updatedAt = curdate() where id = old.orderId;*

*end $$*

*delimiter ;*

*insert into order\_item(orderId, itemId, discount, quantity, createdAt) values*

*(100, 1, 0, 2, curdate()),(100, 2, 0, 1, curdate()),(100, 3, 0, 2, curdate()),*

*(101, 3, 0, 2, curdate()),(101, 3, 0, 1, curdate()),(101, 3, 0, 1, curdate()),*

*(102, 1, 0, 1, curdate()),(102, 2, 0, 2, curdate()),(102, 3, 0, 1, curdate()),*

*(103, 4, 0, 2, curdate()),(103, 2, 0, 1, curdate()),(103, 2, 0, 2, curdate()),*

*(104, 13, 0, 3, curdate()),(104, 7, 0, 2, curdate()),(104, 10, 0, 2, curdate());*

*delete from order\_item where id = 3;*

*update order\_item set quantity = 3 where id = 2;*

*insert into `transaction`(userId, orderId, `code`, `type`, `mode`, `status`, createdAt) values*

*(1012, 100, 'xxx-xx', 'Debit', 'Offline','Success', curdate()),(1007, 101, 'xxx-xx', 'Credit', 'Offline','Success', curdate()),*

*(1008, 102, 'xxx-xx', 'Debit', 'Offline','Success', curdate()),(1009, 103, 'xxx-xx', 'Credit', 'Offline','Success', curdate()),*

*(1010, 104, 'xxx-xx', 'Debit', 'Offline','Success', curdate()),(1011, 105, 'xxx-xx', 'Credit', 'Offline','Cancelled', curdate()),*

*(1012, 106, 'xxx-xx', 'Debit', 'Offline','Cancelled', curdate()),(1007, 107, 'xxx-xx', 'Credit', 'Online','Failed', curdate()),*

*(1011, 108, 'xxx-xx', 'Debit', 'Offline','Cancelled', curdate()),(1008, 109, 'xxx-xx', 'Credit', 'Online','Failed', curdate()),*

*(1008, 110, 'xxx-xx', 'Credit', 'Online','Cancelled', curdate());*

**Part 2 (12 Points):**

* Design and clearly explain **at least ten** query scenarios that may be useful on your database. You may include the scenarios or questions you described in Step 1 question#4.
* Write SQL syntax to answer each query. A query like this: select count(\*) from tableOne, can only be used as one of the ten.
* Strongly suggested to include some queries using Table Joins, Aggregates (at least 2), Group By, Order By, and Having (In other words have a good variety that incorporates the material we have been learning).
* At least one query must use a view (that you created from your application tables) in the FROM clause.

Note: the SELECT used for in the creation of the view, does not count as one of the 10 queries.

*-- -- Scenarios #1 show menu all item.*

*select \* from item;*

*-- -- Scenarios #2 show menu type : Main.*

*select I.id, I.title, I.type, I.price, M.type from item I*

*join (select \* from menu where `type`='Main') M where I.`type` = M.title;*

*-- -- Scenarios #3 show menu type : Drink.*

*select I.id, I.title, I.type, I.price from item I*

*join (select \* from menu where `type`='Drink') M where I.`type` = M.title;*

*-- -- Scenarios #4 show menu type : Alcohol.*

*select I.id, I.title, I.type, I.price from item I*

*join (select \* from menu where `type`='Alcohol') M where I.`type` = M.title;*

*-- -- Scenarios #5 show menu type : Dessert.*

*select I.id, I.title, I.type, I.price from item I*

*join (select \* from menu where `type`='Dessert') M where I.`type` = M.title;*

*-- -- Scenarios #6 show all Restaurant's employee*

*-- select \* from `user` where user\_role = 'employee' or user\_role ='chef';*

*-- -- Scenarios #7 show all Restaurant's customers*

*select \* from `user` where user\_role = 'customer';*

*-- -- Scenarios from Project Handle Step1.*

*-- -- 1. List names and order’s date-time of all customers for the day.*

*select C.fName, C.lName, O.id, O.grandTotal, O.createdAt from `order` O, `user` C*

*where O.userId = C.id and O.createdAt = '2020-11-03';*

*-- -- 2. List out the foods menu which price order by high to low.*

*select \* from item order by price;*

*-- -- 3. List customers name who make a payment more then $50.*

*select C.fName, C.lName, O.grandTotal from `user` C, `order` O where O.userId = C.id and O.grandTotal > 50;*

*-- -- 4. List out the top 5 of popular foods for this week.*

*select I.id 'Food id', I.title, OI.q 'Quantity' from item I,*

*(select OI.itemId i, sum(OI.quantity) q from order\_item OI*

*where createdAt between '2020-11-03' and '2020-11-08' group by OI.itemId having q > 0) OI*

*where I.id = OI.i order by OI.q desc limit 5;*

*-- -- 5. List out all menu which no spicy.*

*drop view if exists `No\_Spicy\_Foods\_Menu`;*

*create view `No\_Spicy\_Foods\_Menu` as select \* from item where content = 'No Spicy';*