CPSC 304 Project Cover Page

Milestone #: __2____

Date: __Oct 24, 2021__

Group Number: ____76____

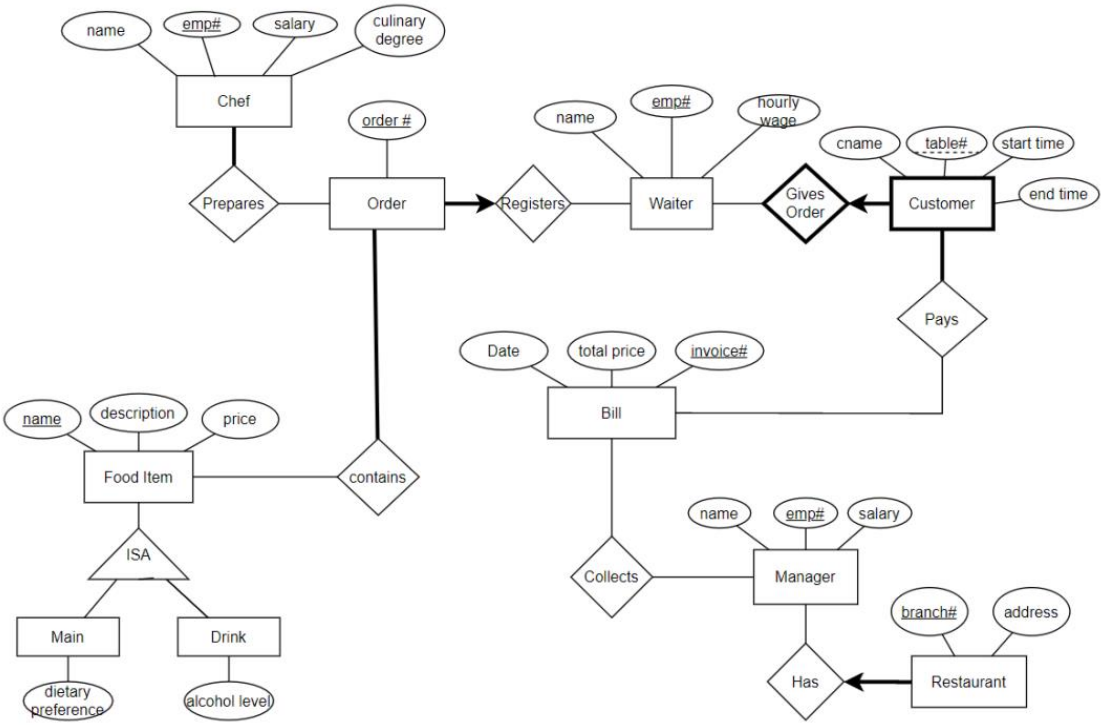
Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Timothy Chui	35908152	v3g3b	timothychui2015@gmail.com
Sunwoo Kim	33494155	c1n1b	Wayne_orange@hotmail.com
Ysabelle Lacasandile	82368176	t5i2b	Y.lacasandile@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

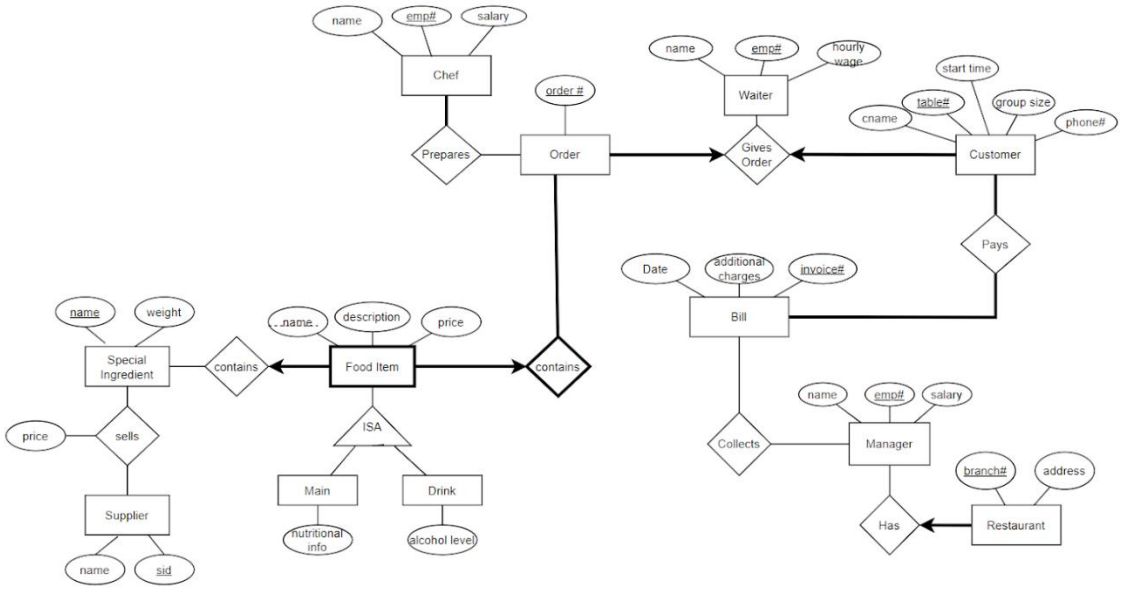
In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Part 2

Milestone 1 ER



Milestone 2 updated ER



Changes	**Reason**
Added attributes to customer	For FD decomposition in milestone 2
Moved the weak entity from “customer Gives Order” to “FoodItem and Order”	The original weak entity did not make sense
Made waiter and customer into a ternary relationship	Better representation of the 3 entities
Added supplier sells special ingredients entity-relationship set	Not enough entity/relationships
Updated some constraints	To allow accurate representation of our model

Part 3

Note: All candidate keys are primary keys

Supplier(sid: int, name: char)
sells(**sid: int**, **ingredName: char**, price: int)
Special Ingredient(ingredName, weight)
containsFoodItem(foodName: char, description: char, price: int, **IngredName: char, order#: int**)
(IngredName is not null)
Main(name: char, nutritInfo: char)
Drinks(name: char, alcoholLv: int)
hasRestaurant(branch#: int, **emp#: int,** streetNum: char, streetName:char, city: char, country: char, postalCode: char)
Manager(name: char, emp#: int, salary: int)
Collects(**emp#: int, invoice#: int**)
Bill(date: char, addCharges: int, invoice#: int)
Pays(**invoice#: int, table#: int**) (need assertion)
Customer(cname: char, table#: int, startTime: int. groupSize: int, phone#: int)
givesOrderOrder(order#: int, **emp#: int, table#: int**) (table# is unique)
Waiter(name: char, emp#: int, wage: int)
Prepares(**order#: int, emp#: int**)
Chef(name: char, emp#: int, salary: int)

Part 4

Supplier(sid, name)
Sid -> name

sells(**sid**, **ingredName**, price)
Sid, ingredName -> price

Special Ingredient(ingredName, weight)
ingredName -> price

containsFoodItem(foodName, description, price, **IngredName, order#**)
foodName, order# -> description, price, IngredName
foodName -> description
Description -> IngredName

Main(name, nutritInfo)
Name -> nutritInfo

Drinks(name, alcoholLv)
Name -> alcoholLv

hasRestaurant(branch#, **emp#,** address)
branch# -> emp#, address

Manager(name, emp#, salary)
emp# -> name, salary

Collects(**emp#, invoice#**)

Bill(date, addCharges, invoice#)
invoice# -> date, addCharges

Pays(**invoice#, table#**) (need assertion)

Customer(table#, cname, startTime. groupSize, phone#)
table# -> cname, startTime. groupSize, phone#
Cname, phone# -> groupSize, startTime

givesOrderOrder(order#, **emp#, table#**) (table# is unique)
order# -> emp#, table#

Waiter(name, emp#, wage)

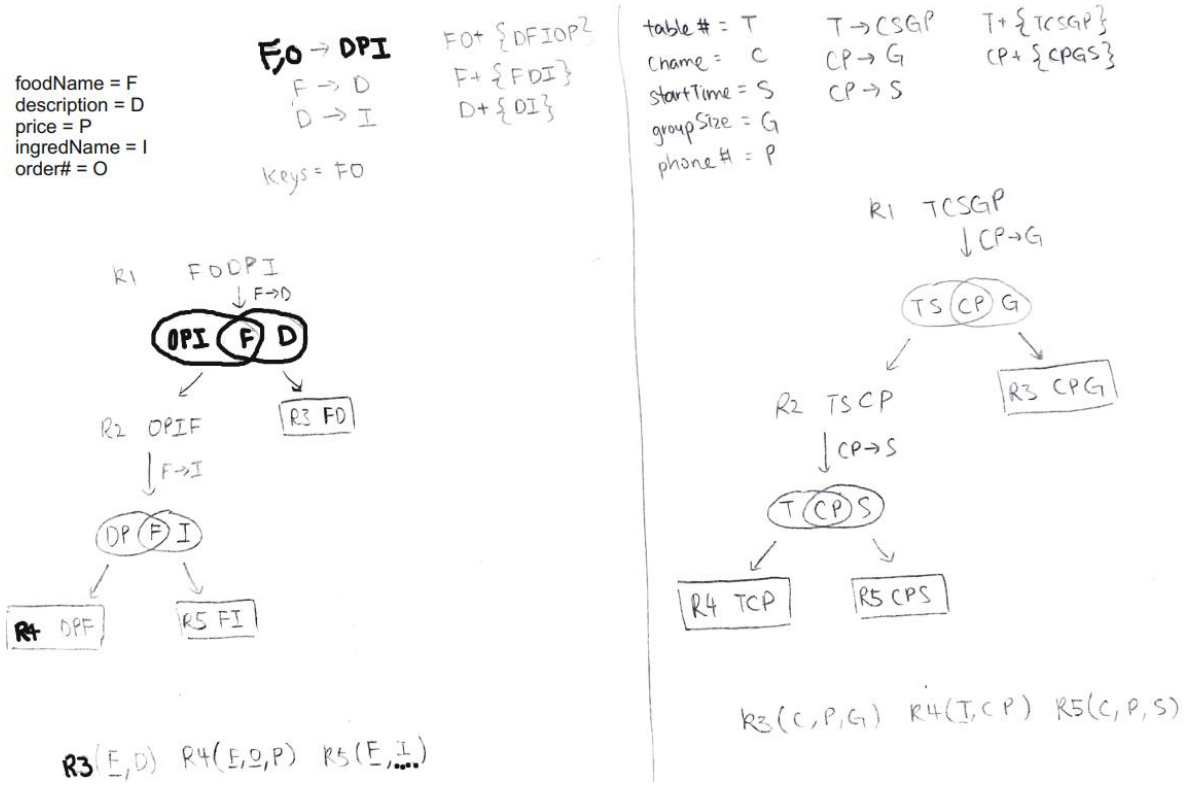
Emp# -> name, wage

Prepares(**order#, emp#**)

Chef(name, emp#, salary)
emp# -> name, salary

Part 5

Decomposed table:



containsFoodItem(<u>foodName</u>, description, price, **IngredName, <u>order#</u>**) becomes:
R1(<u>foodName</u>, description)
R2(<u>foodName</u>, price**, <u>order#</u>**)
R3(f<u>oodName</u>, **IngredName**)

Customer(<u>table#</u>, cname, startTime. groupSize, phone#) becomes:
R1(cname. groupSize, phone#)
R2(<u>table#</u>, cname, phone#)
R3(cname, startTime, phone#)

Part 6

```
CREATE TABLE Supplier (
        sid INT PRIMARY KEY,
        name CHAR);

CREATE TABLE Sells (
        sid INT,
        ingredName INT,
        price INT,
        PRIMARY KEY (sid, ingredName),
        FOREIGN KEY (sid) REFERENCES Supplier,
        FOREIGN KEY (ingredName) REFERENCES Special Ingredient);

CREATE TABLE Special Special Ingredient(
        ingredName CHAR PRIMARY KEY,
        weight INT);

CREATE TABLE containsFoodItem (
        foodName CHAR,
        description CHAR,
        price INT,
        ingredName CHAR NOT NULL,
        order# INT,
        PRIMARY KEY (order#, foodName),
        FOREIGN KEY (order#) REFERENCES givesOrderOrder
ON DELETE CASCADE,
        FOREIGN KEY (ingredName) REFERENCES Ingredient);

CREATE TABLE Main (
        name CHAR PRIMARY KEY,
        nutritInfo CHAR);

CREATE TABLE Drink (
        name CHAR PRIMARY KEY,
        alcoholLv INT);

CREATE TABLE hasRestaurant (
        branch# INT PRIMARY KEY,
address CHAR,
emp# INT NOT NULL,
FOREIGN KEY (emp#) REFERENCES Manager);
```

```
CREATE TABLE Manager (
        name CHAR,
        emp# INT PRIMARY KEY,
salary INT);

CREATE TABLE Collects (
        emp# INT,
invoice# INT,
PRIMARY KEY (emp#, invoice#),
FOREIGN KEY (emp#) REFERENCES Manager,
FOREIGN KEY (invoice#) REFERENCES Bill);

CREATE TABLE Bill (
        date CHAR,
        addCharges INT,
        invoice# INT PRIMARY KEY);

CREATE TABLE Pays (
        invoice# INT,
        table# INT,
        PRIMARY KEY (invoice#, table#),
        FOREIGN KEY (invoice#) REFERENCES Bill,
        FOREIGN KEY (table#) REFERENCES Customer);

CREATE TABLE Customer (
        cname CHAR,
        table# INT PRIMARY KEY);

CREATE TABLE givesOrderOrder (
        order# INT PRIMARY KEY,
        emp# INT,
        table# INT,
        FOREIGN KEY (emp#) REFERENCES Waiter,
        UNIQUE table#);

CREATE TABLE Waiter (
        name CHAR,
        emp# INT PRIMARY KEY,
        wage INT);

CREATE TABLE Prepares (
        order# INT,
        emp# INT,
        PRIMARY KEY (order#, emp#)
```

```
        FOREIGN KEY (order#) REFERENCES givesOrderOrder,
        FOREIGN KEY (emp#) REFERENCES Chef);

CREATE TABLE Chef (
        name CHAR,
        emp# INT PRIMARY KEY,
        salary INT);
```

Part 7

Supplier	
sid	name
1234	Bob's Foods
1235	Not Bob's Foods
1236	Fred's Fish
1237	We Sell Cow
1238	Vegetables Inc.

Manager		
name	emp#	salary
Jim	1233	4000
Jim	1233	4000
Jim	1233	4000
Jim	1233	4000
Jim	1233	4000

preparesOrder	
order#	emp#
112	1245
113	1245
114	1245
115	1245
116	1245

Sells		
sid	ingredName	price
1234	Chicken Wings	199.99
1234	Pork Chops	249.99
1236	Salmon	329.49
1237	Ribeye	399.99
1238	Broccoli	429.19

Main	
name	nutritInfo
Avocado Burger	5000 calories
Fettuccine	150 calories
cream pie	gluten free
garlic bread	protein free
pizza	gluten free

Collects	
emp#	invoice#
1233	1
1233	2
1233	3
1233	4
1233	5

Special Ingredient	
ingredName	weight
Chicken Wings	50
Pork Chops	40
Salmon	60
Ribeye	50
Broccoli	80

Drink	
name	alcoholLv
lady mary	8
gin gin	20
apple juice	0
coke	0
tiger bomb	12

Pays	
invoice#	table#
1	100
2	200
3	300
4	400
5	500

Waiter		
name	emp#	wage
Da	1433	15
De	1444	13
Di	1455	15
Do	1466	14
Du	1477	15

Prepares	
order#	emp#
112	1333
113	1344
114	1355
115	1366
116	1377

Chef		
name	emp#	salary
Ca	1333	2000
Ce	1344	3000
Ci	1355	4000
Co	1366	3500
Cu	1377	2500

Bill		
date	addCharges	invoice#
Oct 11	20	1
10/11/2021	30	2
10/11/2021	20	3
Oct 11	2	4
Oct 12	20	5

givesOrderOrder		
order#	emp#	table#
112	1433	100
113	1433	200
114	1433	300
115	1466	400
116	1477	500

hasRestaurant		
branch#	address	emp#
11	happy rd	1233
22	unhappy rd	1233
33	sad rd	1233
44	excited ave	1233
55	angry rd	1233

Customer R1		
cname	groupSize	phone#
Bob	2	1234567890
Bobby	2	2345678901
Bobbette	3	3456789012
Bobnathon	4	4567890123
Bobu	5	5678901234

Customer R2		
table#	cname	phone#
100	Bob	1234567890
200	Bobby	2345678901
300	Bobbette	3456789012
400	Bobnathon	4567890123
500	Bobu	5678901234

Customer R3		
cname	startTime	phone#
Bob	9:00 AM	1234567890
Bobby	9:00 AM	2345678901
Bobbette	9:00 AM	3456789012
Bobnathon	9:00 AM	4567890123
Bobu	10:00 AM	5678901234

containsFoodItemR1

foodName	description
bob wings	it has chicken
Steak	From a pig
baked salmon	fresh salmon
ribeye steak	it's a cow
Jim's surprise	salad
ribeye steak	it's a cow

containsFoodItemR2

foodName	price	order#
bob wings	10	112
Steak	20	113
baked salmon	30	114
ribeye steak	20	115
Jim's surprise	10	116
ribeye steak	20	116

containsFoodItemR3

foodName	ingredName
bob wings	Chicken Wings
Steak	Pork Chops
baked salmon	Salmon
ribeye steak	Ribeye
Jim's surprise	Broccoli
ribeye steak	Ribeye

Before Decomposition

Customer

table#	cname	start time	group size	phone#
100	Bob	9:00 AM	2	1234567890
200	Bobby	9:00 AM	2	2345678901
300	Bobbette	9:00 AM	3	3456789012
400	Bobnathon	9:00 AM	4	4567890123
500	Bobu	10:00 AM	5	5678901234

containsFoodItem

foodName	description	price	ingredName	order#
bob wings	it has chicken	1	Chicken Wings	112
Granny ham	From a pig	2	Pork Chops	113
baked salmon	fresh salmon	3	Salmon	114
ribeye steak	it's a cow	4	Ribeye	115
Jim's surprise	salad	5	Broccoli	116

Part 8

Supplier
Insertion: add supplier to the supplier table
Delete: delete supplier from supplier table when they can no longer provide ingredients

Manager
Insertion: add newly hired manager to the manager table
Update: update manager's salary in the manager's table if given a raise
Delete: delete manager from manager's table when they are no longer working at the restaurant

Sells
Insertion: add ingredient name in the sells table
Update: update price in sells table if supplier increases their selling price
Delete: delete ingredient in the sells table when we no longer need it for food item

Special Ingredient
Insertion: add weight in special ingredient table
Update: update weight in special ingredient table if changing the portions
Delete: delete ingredient name in ingredient table when no longer using that ingredient for food item

Main
Insertion: add nutritional info in the main table
Update: update nutritional info when ingredients in main has changed
Delete: delete nutritional info in main table when no longer serving that dish

Bill
Insertion: add invoice number in the bill table
Update: update additional charge in the bill table when customer orders more food item

givesOrderOrder
Insertion: add table# in givesOrderOrder table when customer gives their order
Delete: delete order number in givesOrderOrder table when Customer has finished eating and left the restaurant

We have not yet covered the other queries in class.