Database Concepts CGS 2545 Section 001 SPRING 2017

Group 18 Project Phase 4

Group Members: Santiago Perez A, Justin Kardach, Kyle T, Lou C

Table of Contents

Phase 0 Report

Restaurants Near UCF

Proposal/Description: The database is a record of of all the restaurants near UCF. Our database will contain the names of restaurants, service offered, types of food, and the location. Our database will classify restaurants to facilitate search of food options. The purpose of the database is to allow students to find desired food options near campus.

1. UoD: All restaurants within five miles from the University of Central Florida
2. Business Rules
   1. Each establishment that sells prepared/cooked food is a restaurant.
   2. The manner in which each restaurant sells food is a service.
   3. Any person that purchases food from a restaurant is a customer.
   4. Food is any prepared meal served at a restaurant.
   5. Each restaurant has a name.
   6. Each restaurant has a unique address.
   7. Price is the range of money spent on an item of food at a restaurant.
   8. Each service has a type
   9. Each food has a category
3. Entities and Relationships
   1. Restaurant
   2. Customers
   3. Service
   4. Food
   5. Price

Customers eats Food

Restaurant sells Food

Service provided by Restaurant

Food has Price

Restaurant provides service

Phase 1 Report

Between our Phase 0 Proposal Phase and Phase 1, more business rules/constraints were added and our ER diagram was created.

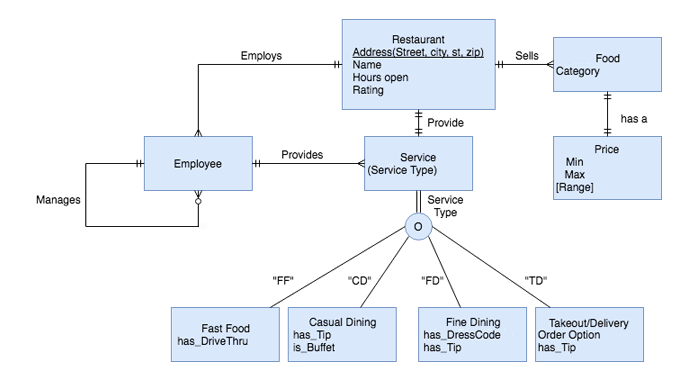
Business Rules:

1. Each establishment that sells prepared/cooked food is a restaurant.
2. Each restaurant sells food.
3. Each restaurant has a name, a unique address, hours open, and rating
4. The address is a composite identifier with the components, street number, street, and zip
5. Each restaurant has a service.
6. Each restaurant employs employees
7. Food is any prepared meal served at a restaurant.
8. Each food has a category
9. Each food has price range.
10. Price range has the following attributes, min, max, and a derived attribute, range.
11. The manner in which each restaurant sells food is a service.
12. An employee provides a service.
13. Any person that purchases food from a restaurant is a customer.
14. Price is the range of money spent on an item of food at a restaurant.
15. Each food has a category.
16. An employee is managed by one other employee.
17. An employee can manage one, zero, or many employees.
18. A restaurant can sell many foods.
19. A service has to be at least fast food, casual dining, buffet, take out/online, or fine dining. A service can be more than one type.

Service

1. Any restaurant that sells and provides menu items in under 10 minutes is Fast food. Fast food has the attribute has\_DriveThru.
2. Any restaurant that follows a meal structure is Casual dining. Casual dining has the attributes has\_Tip and is\_Buffet
3. Any restaurant that allows ordering menu items from online or by phone is Take out/Delivery. Takeout/Delivery has the attributes Order Option and has\_Tip.
4. Any restaurant that follows a meal structure and dress code is considered fine dining. Fine Dining has the attributes has\_DressCode and has\_Tip.

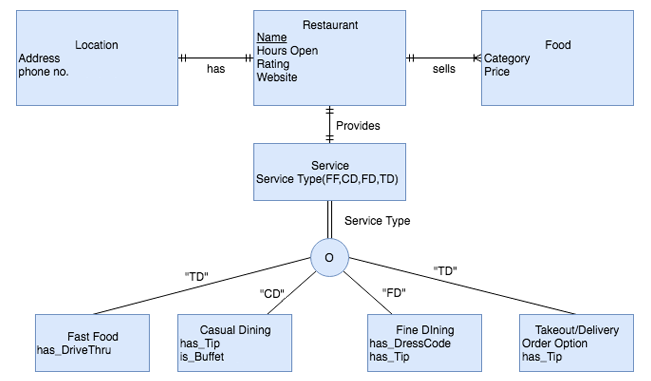
ER Diagram

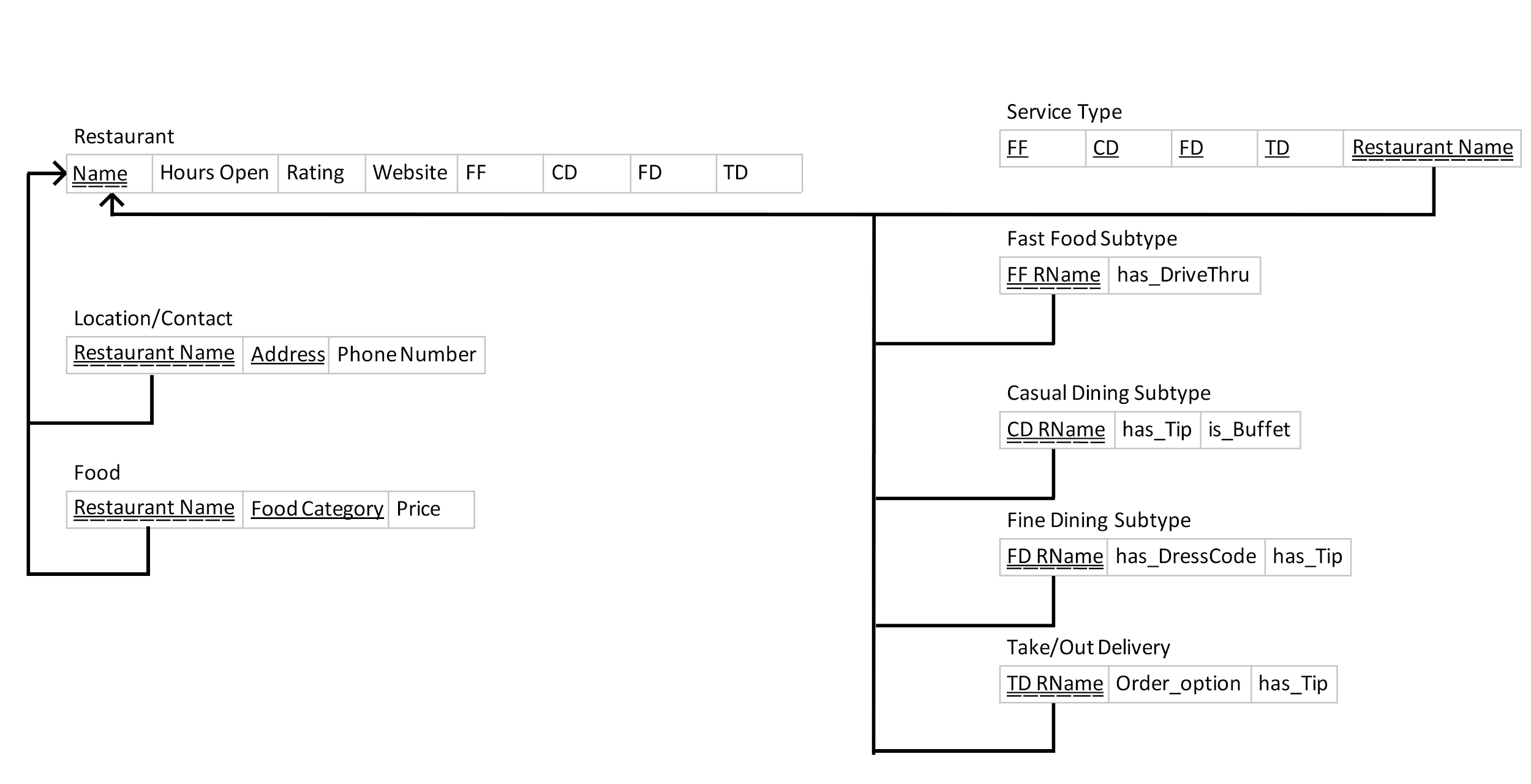


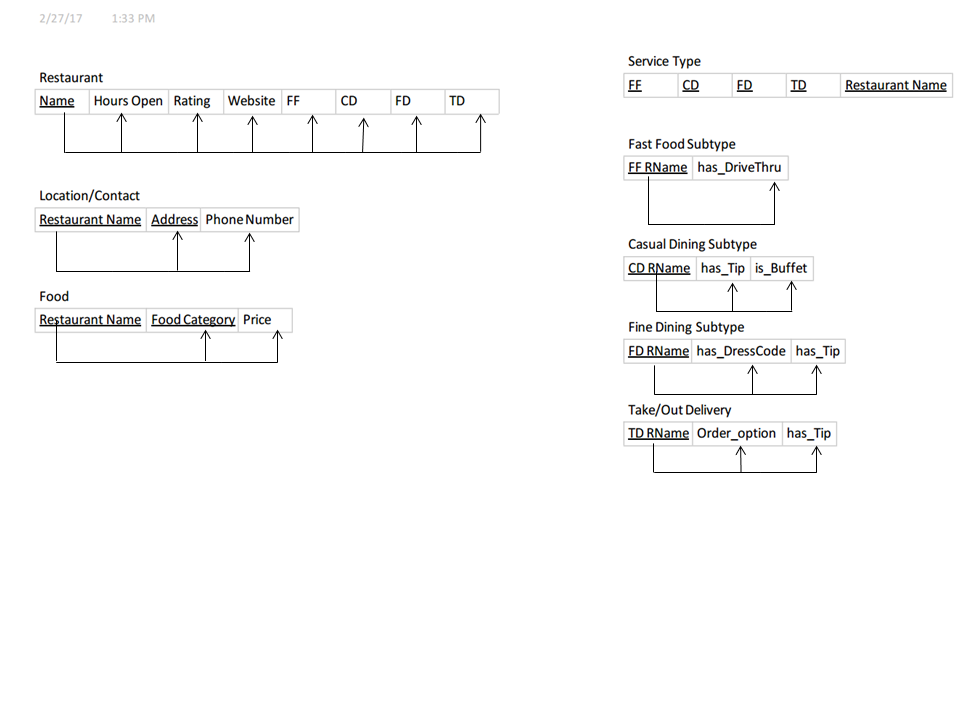
Phase 2 Report

Please note that between the Phase 1 and Phase 2 submission, we eliminated both the Employee and Price entities. Price became an attribute under the Food entity, and the Location entity was created to get rid of the composite attribute under Restaurant. The elimination and creation of these entities caused us to change/modify many of our business rules to maintain the integrity of our database. Under the functional dependencies shown for the schemas, the arrows were not added under Service Type. This was fixed in the Phase 3 submission.

ER Diagram



Schemas



Location/Contact: The schema has no multivalued attributes. Therefore, it is 1NF. The keys “Restaurant Name” and “Address” form a full functional dependency with “Phone Number” and there are no partial functional dependencies. Therefore it is 2NF. Since it is 1NF, 2NF, and “Phone Number” is the only non-key attribute it is 3NF because no transitive functional dependencies are possible.

Restaurant: The schema has no multivalued attributes. Therefore, it is 1NF. All of the non key attributes, “Hours Open”, “Rating”, and “Website” are consequents of the primary key, “Name”. “FF”, “CD”, “FD”, and “TD” are also determined by the primary key “Name”. Because are no partial functional dependencies or transitive functional dependencies, it is 2NF and 3NF.

Food: The schema has no multivalued attributes. Therefore, it is 1NF. The keys “Restaurant Name” and “Food Category” form a full functional dependency with “Price” and there are no partial functional dependencies. Therefore it is 2NF. Since it is 1NF, 2NF, and “Price” is the only non-key attribute it is 3NF because no transitive functional dependencies are possible.

Service: Service does not have any multivalued attributes, therefore it is 1NF. It also does not have any partial functional dependencies, which makes it 2NF. Service does not have any transitive functional dependencies, which means it is also 3NF.

Fast Food: Since the table has no multivalued attributes and the keys “FF” and “FastFood RName” form a full functional dependency with “has\_Drivethru”, it is 1NF and 2NF. Since “has\_Drivethru” is the only non-key attribute, no transitive dependencies are possible. It is therefore, 3NF.

Casual Dining: Casual Dining does not have any multivalued attributes, therefore it is 1NF. It also does not have any partial functional dependencies, which makes it 2NF. Casual Dining does not have any transitive functional dependencies, which means it is also 3NF. The composite primary key, “CD” and “Casual RName”, determine the attributes “has\_Tip”, and “is\_Buffet”.

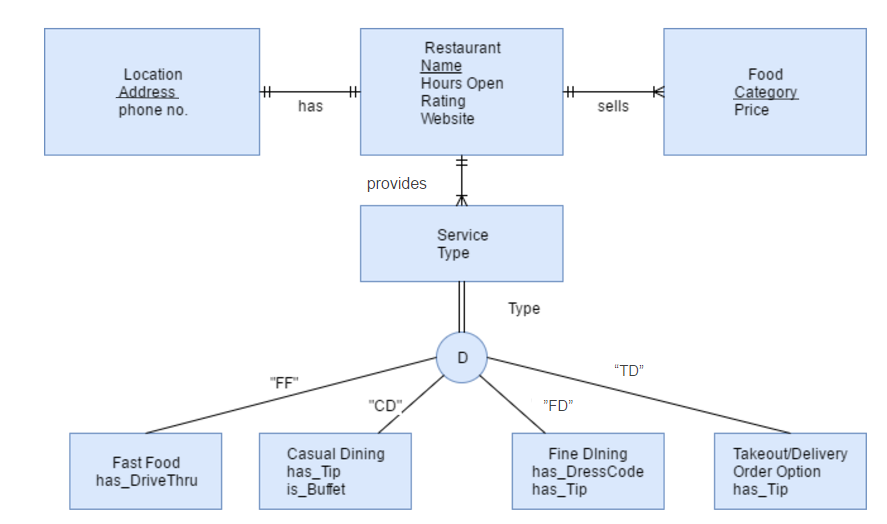
Fine Dining: Fine Dining does not have any multivalued attributes, partial functional dependencies, or transitive functional dependencies, which makes Fine Dining 1NF, 2NF, and 3NF. The full functional dependencies present are “FD” and “Fine RName” determine “has\_DressCode”, and “has\_Tip”.

TakeOut/Delivery: TakeOut/Delivery is in 1NF because it does not have any multivalued attributes. It is also in 2NF because it does not have any partial functional dependencies. It is in 3NF because it does not have transitive functional dependencies. The composite primary key, “TD” and “TD RName”, determine the attributes “Order Option” and “has\_Tip”.

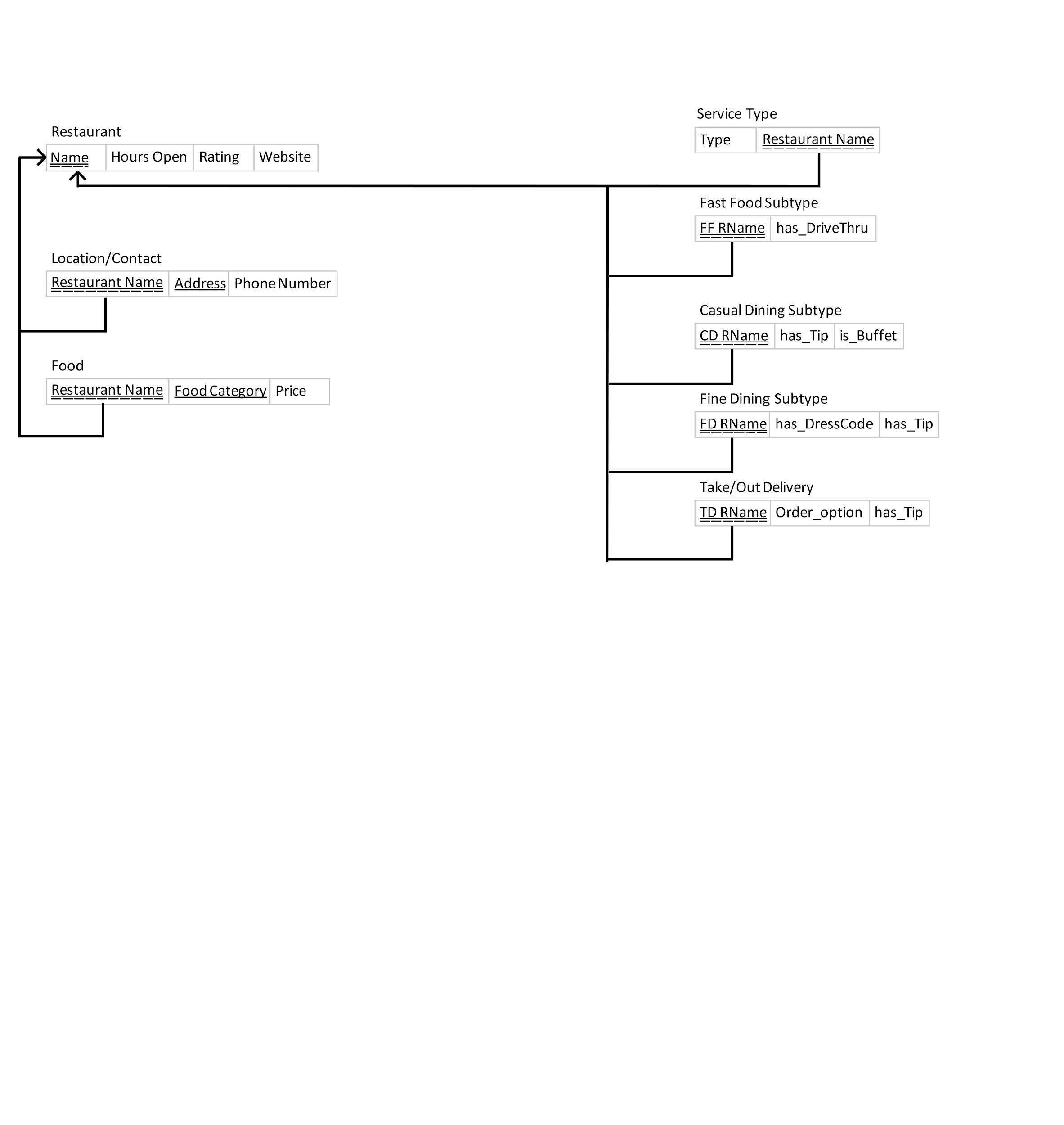
Phase 3 Report

Please note that between Phase 2 and Phase 3 some changes were made on the ER diagram and Schemas. The Restaurant entity can now have one or many Services, while Service became a Disjointed Subtype instead of an Overlap. An updated schema was included as well as the script files used to populate/create our database. Example scripts are shown towards the bottom with screenshots of returned results.

ER Diagram



Schemas



**#Database Creation Script**

# Group: 18

# Names: Santiago Perez Arrubla, Justin Kardach, Kyle Turner, Lou Carlos

# Date: 4/4/17

# ===============================================

# Creation Script of the tables in the database.

# Creates table for the restaurant.

# Contains the following attributes: Name, Hours\_Open, Rating, Website.

# Primay key = Name

CREATE TABLE Restaurant (

Name varchar(30) not null,

Hours\_Open varchar(30),

Rating Integer,

Website varchar(30),

CONSTRAINT RestaurantPK PRIMARY KEY(Name));

# Creates table for the service type.

# Contains the following attributes: RestaurantName, TYPE

# Type must be FF = FastFood, CD = Casual Dining, FD = Fine Dining, TD = Takeout/Delivery

# Primary key = RestaurantName & Type (Composite Key)

# Foreign key = Name from Restaurant Table.

CREATE TABLE ServiceType (

RestaurantName varchar(30) not null,

Type varchar(2),

check(Type in(“FF”, “CD”, “FD”, “TD”)),

CONSTRAINT ServiceTypePK PRIMARY KEY(RestaurantName, Type),

CONSTRAINT ServiceTypeFK FOREIGN KEY(RestaurantName) REFERENCES Restaurant(Name));

# Creates table for Contact.

# Contains the following attributes: RestaurantName, Address, PhoneNumber.

# Primary key = RestaurantName & Address. (Composite Key)

# Foreign key = Name from Restaurant Table.

CREATE TABLE Contact (

RestaurantName varchar(30) not null,

Address varchar(30) not null,

PhoneNumber numeric(10,0),

CONSTRAINT ContactPK PRIMARY KEY(RestaurantName, Address),

CONSTRAINT ContactFK FOREIGN KEY(RestaurantName) REFERENCES Restaurant(Name));

# Creates table for Food.

# Contains the following attributes: RestaurantName, FoodCategory, AVGPrice.

# FoodCategory must be Latin, African, Asian, American, European, Fusion, Australian, Antarctican, or Other.

# Primary key = RestaurantName & FoodCategory. (Composite Key)

# Foreign key = Name from Restaurant Table.

CREATE TABLE Food (

RestaurantName varchar(30),

FoodCategory varchar(30),

check(FoodCategory in('Latin', 'African', 'Asian', 'American', 'European', 'Fusion', 'Australian', 'Antarctican', 'Other')),

AVGPrice integer,

CONSTRAINT FoodPK PRIMARY KEY(RestaurantName, FoodCategory),

CONSTRAINT FoodFK FOREIGN KEY(RestaurantName) REFERENCES Restaurant(Name));

# Creates table for FF\_Sub.

# Contains the following attributes: FF\_RName, hasDriveThru.

# Primary key = FF\_RName.

# Foreign key = FF\_RName from Restaurant Table.

CREATE TABLE FF\_Sub (

FF\_RName varchar(30),

hasDriveThru boolean,

CONSTRAINT FF\_SubPK PRIMARY KEY(FF\_Rname),

CONSTRAINT FF\_SubFK FOREIGN KEY(FF\_RName) REFERENCES Restaurant(Name));

# Creates table for CD\_Sub.

# Contains the following attributes: CD\_RName, hasTip, isBuffet.

# Primary key = CD\_RName.

# Foreign key = CD\_RName from Restaurant Table.

CREATE TABLE CD\_Sub (

CD\_RName varchar(30),

hasTip boolean,

isBuffet boolean,

CONSTRAINT CD\_SubPK PRIMARY KEY(CD\_RName),

CONSTRAINT CD\_SubFK FOREIGN KEY(CD\_RName) REFERENCES Restaurant(Name));

# Creates table for FD\_Sub.

# Contains the following attributes: FD\_RName, hasTip, hasDressCode.

# Primary key = FD\_RName.

# Foreign key = FD\_RName from Restaurant Table.

CREATE TABLE FD\_Sub (

FD\_RName varchar(30),

hasTip boolean,

hasDressCode boolean,

CONSTRAINT FD\_SubPK PRIMARY KEY(FD\_RName),

CONSTRAINT FD\_SubFK FOREIGN KEY(FD\_RName) REFERENCES Restaurant(Name));

# Creates table for TD\_Sub.

# Contains the following attributes: FD\_RName, orderOption, hasTip.

# orderOption must be: Phone, Internet, UberEats, Delivery, Takeout.

# Primary key = TD\_RName.

# Foreign key = TD\_RName from Restaurant Table.

CREATE TABLE TD\_Sub (

TD\_RName varchar(30),

orderOption varchar(30),

check(Type in('Phone', 'Internet', 'UberEats', 'Delivery', 'Take Out')),

hasTip boolean,

CONSTRAINT TD\_SubPK PRIMARY KEY(TD\_RName),

CONSTRAINT TD\_SubFK FOREIGN KEY(TD\_RName) REFERENCES Restaurant(Name));

**#Database Population Script**

# Group: 18

# Names: Santiago Perez Arrubla, Justin Kardach, Kyle Turner, Lou Carlos

# Date: 4/4/17

# ===============================================

# Populates the tables in the database

# Populates the restaurant table.

INSERT INTO Restaurant VALUES('Qdoba', '10AM-9PM', 4, 'qdoba.com');

INSERT INTO Restaurant VALUES('Chick-fil-a', '730AM-8pm', 4, 'chick-fil-a.com');

INSERT INTO Restaurant VALUES('Dominos Pizza', '10AM-10PM', 3, 'dominos.com');

INSERT INTO Restaurant VALUES('Dunkin Donuts', '7AM-12AM', 5, 'dunkindonuts.com');

INSERT INTO Restaurant VALUES('Burger U', '11AM-11PM', 5, 'burgeruucf.com');

INSERT INTO Restaurant VALUES('Moms Spaghetti', '9AM-5PM', 3, 'spaghetti.com');

INSERT INTO Restaurant VALUES('McWendys', '8AM-7PM', 4, 'mcwendys.com');

INSERT INTO Restaurant VALUES('Burger Queen', '8AM-2AM', 2, 'burgerqueen.com');

INSERT INTO Restaurant VALUES('Taco City', '7AM-3AM', 1, 'tacocity.com');

INSERT INTO Restaurant VALUES('Tijuana Fats', '10AM-11PM', 3, 'tfats.com');

INSERT INTO Restaurant VALUES('Pizza Bell', '8AM-5AM', 2, 'pizzabell.com');

INSERT INTO Restaurant VALUES('Bwekfast', '330AM-420AM', 3, 'bwekfast.com');

INSERT INTO Restaurant VALUES('Launch', '1030AM-5PM', 4, 'launch.com');

INSERT INTO Restaurant VALUES('Danner', '530PM-1AM', 5, 'danner.com');

INSERT INTO Restaurant VALUES('Jebaited', '11AM-11PM', 4, 'jebaited.com');

INSERT INTO Restaurant VALUES('RIP in Pepperonis', '10AM-1AM', 5, 'ripinpepperonies.com');

INSERT INTO Restaurant VALUES('iDennys', '7AM-5PM', 2, 'idennys.com');

INSERT INTO Restaurant VALUES('Subhue', '8AM-10PM', 5, 'huehuehue.com');

INSERT INTO Restaurant VALUES('Wendalds', '7AM-7PM', 3, 'wendalds.com');

INSERT INTO Restaurant VALUES('Elvas', '1230PM-120pm', 5, 'elvas.edu');

# Populates the ServiceType table.

INSERT INTO ServiceType VALUES('Qdoba', 'FF');

INSERT INTO ServiceType VALUES('Chick-fil-a', 'FF');

INSERT INTO ServiceType VALUES('Dominos Pizza', 'TD');

INSERT INTO ServiceType VALUES('Dunkin Donuts', 'FF');

INSERT INTO ServiceType VALUES('Burger U', 'CD');

INSERT INTO ServiceType VALUES('Moms Spaghetti', 'CD');

INSERT INTO ServiceType VALUES('McWendys', 'FF');

INSERT INTO ServiceType VALUES('Burger Queen', 'FF');

INSERT INTO ServiceType VALUES('Taco City', 'TD');

INSERT INTO ServiceType VALUES('Tijuana Fats', 'CD');

INSERT INTO ServiceType VALUES('Pizza Bell', 'TD');

INSERT INTO ServiceType VALUES('Bwekfast', 'CD');

INSERT INTO ServiceType VALUES('Launch', 'CD');

INSERT INTO ServiceType VALUES('Danner', 'FD');

INSERT INTO ServiceType VALUES('Jebaited', 'FD');

INSERT INTO ServiceType VALUES('RIP in Pepperonis', 'TD');

INSERT INTO ServiceType VALUES('iDennys', 'CD');

INSERT INTO ServiceType VALUES('Subhue', 'CD');

INSERT INTO ServiceType VALUES('Wendalds', 'FF');

INSERT INTO ServiceType VALUES('Elvas', 'FD');

# Populates the Contact table.

INSERT INTO Contact VALUES('Qdoba', 'CD12715 Pegasus Dr', 4072753820);

INSERT INTO Contact VALUES('Chick-fil-a','4000 Central Florida Blvd', 4078232238);

INSERT INTO Contact VALUES('Dominos Pizza','4000 Central Florida Blvd', 4078232238);

INSERT INTO Contact VALUES('Dunkin Donuts','4210 W Plaza Dr', 4078823688);

INSERT INTO Contact VALUES('Burger U','4233 E Plaza Dr', 4078820450);

INSERT INTO Contact VALUES('Moms Spaghetti', '8 Mile Rd', 4071234567);

INSERT INTO Contact VALUES('McWendys', '1 Yellow Brick Rd', 4073219654);

INSERT INTO Contact VALUES('Burger Queen', '123 Fake St', 4074543678);

INSERT INTO Contact VALUES('Taco City', '537 Candyland Way', 4074516782);

INSERT INTO Contact VALUES('Tijuana Fats', '42 Wallaby Way', 4074516543);

INSERT INTO Contact VALUES('Pizza Bell', '000 Circle Cr', 4074560000);

INSERT INTO Contact VALUES('Bwekfast', '12 Waffles Dr', 4079871235);

INSERT INTO Contact VALUES('Launch', '400 Sandwich Way', 4075679823);

INSERT INTO Contact VALUES('Danner', '300 Pasta Dr', 4075679823);

INSERT INTO Contact VALUES('Jebaited', '1 Central Florida Blvd', 4076787890);

INSERT INTO Contact VALUES('RIP in Pepperonis', '123 Orange Ave', 4079089089);

INSERT INTO Contact VALUES('iDennys', '657 Yellow St', 4076909143);

INSERT INTO Contact VALUES('Subhue', '432 Pepperoni Dr', 4074562345);

INSERT INTO Contact VALUES('Wendalds', '678 Burger St', 4076091832);

INSERT INTO Contact VALUES('Elvas', '125 HEC', 4071233456);

# Populates the Food table.

INSERT INTO Food VALUES('Qdoba', 'Latin', 2);

INSERT INTO Food VALUES('Chick-fil-a', 'Asian', 1);

INSERT INTO Food VALUES('Dominos Pizza', 'Fusion', 1);

INSERT INTO Food VALUES('Dunkin Donuts', 'Antarctican', 1);

INSERT INTO Food VALUES('Burger U', 'European', 2);

INSERT INTO Food VALUES('Moms Spaghetti', 'Antarctican', 3);

INSERT INTO Food VALUES('McWendys', 'Fusion', 0);

INSERT INTO Food VALUES('Burger Queen', 'Fusion', 3);

INSERT INTO Food VALUES('Taco City', 'African', 2);

INSERT INTO Food VALUES('Tijuana Fats', 'American', 3);

INSERT INTO Food VALUES('Pizza Bell', 'Other', 2);

INSERT INTO Food VALUES('Bwekfast', 'American', 1);

INSERT INTO Food VALUES('Launch', 'European', 3);

INSERT INTO Food VALUES('Danner', 'Asian', 2);

INSERT INTO Food VALUES('Jebaited', 'Fusion', 1);

INSERT INTO Food VALUES('RIP in Pepperonis', 'Other', 3);

INSERT INTO Food VALUES('iDennys', 'American',4);

INSERT INTO Food VALUES('Subhue', 'Fusion', 1);

INSERT INTO Food VALUES('Wendalds', 'Other', 2);

INSERT INTO Food VALUES('Elvas', 'Other', 5);

# Populates the FF\_Sub table.

INSERT INTO FF\_Sub VALUES('Qdoba', 0);

INSERT INTO FF\_Sub VALUES ('Chick-fil-a', 0);

INSERT INTO FF\_Sub VALUES('Dunkin Donuts', 0);

INSERT INTO FF\_Sub VALUES ('McWendys', 1);

INSERT INTO FF\_Sub VALUES('Burger Queen', 1);

INSERT INTO FF\_Sub VALUES('Wendalds', 0);

# Populates the CD\_Sub table.

INSERT INTO CD\_Sub VALUES('Burger U', 1, 0);

INSERT INTO CD\_Sub VALUES('Moms Spaghetti' , 0, 1);

INSERT INTO CD\_Sub VALUES('Tijuana Fats', 1, 0);

INSERT INTO CD\_Sub VALUES('Bwekfast', 0, 1);

INSERT INTO CD\_Sub VALUES('Launch', 0, 1);

INSERT INTO CD\_Sub VALUES('iDennys', 1, 0);

INSERT INTO CD\_Sub VALUES('Subhue', 1, 1);

# Populates the FD\_Sub table.

INSERT INTO FD\_Sub VALUES('Danner', 1, 1);

INSERT INTO FD\_Sub VALUES('Jebaited', 1, 1);

INSERT INTO FD\_Sub VALUES('Elvas', 1, 1);

# Populates the TD\_Sub table.

INSERT INTO TD\_Sub VALUES('Dominos Pizza', 'Internet', 1);

INSERT INTO TD\_Sub VALUES('Taco City', 'Phone', 1);

INSERT INTO TD\_Sub VALUES('Pizza Bell', 'Delivery', 1);

INSERT INTO TD\_Sub VALUES('RIP in Pepperonis', 'Take Out', 0);

**#Database Update Script**

# Group: 18

# Names: Santiago Perez Arrubla, Justin Kardach, Kyle Turner, Lou Carlos

# Date: 4/10/17

# ===============================================

# Scripts that show functionality of the database.

#Changes the phone number for the restaurant Qdoba

UPDATE Contact

SET PhoneNumber = 4075433345

WHERE RestaurantName = 'Qdoba';

#Fetches the restaurant name and average price for an American food restaurants from high avg to low

SELECT RestaurantName,AVGPrice

FROM Food

WHERE FoodCategory = 'American'

ORDER BY AVGPrice Desc;

#Fetches the restaurant name and food category for all casual dining restaurants that do not require you to tip.

SELECT RestaurantName, FoodCategory

FROM Food, CD\_Sub

WHERE CD\_Sub.CD\_RName = Food.RestaurantName

AND hasTip = 0;

#Order all casual dining restaurants from high to low rating.

SELECT RestaurantName, Rating

FROM Restaurant, ServiceType

WHERE ServiceType.RestaurantName = Restaurant.Name

ORDER BY Rating DESC;

# Makes sure all foods have a price range

UPDATE Food

SET AVGPrice = 1

WHERE AVGPrice = 0;

# Gets the address of all restaurants with a drive thru

SELECT distinct Contact.RestaurantName, Contact.Address

FROM Contact, FF\_Sub

WHERE Contact.RestaurantName = FF\_Sub.FF\_RName

AND FF\_Sub.hasDriveThru = 1;

# Gets the phone numbers of all restaurants that provide take out/delivery service

SELECT Contact.RestaurantName, Contact.PhoneNumber

FROM Contact, ServiceType

WHERE Contact.RestaurantName = ServiceType.RestaurantName

AND ServiceType.Type = 'TD';

# Gets all the restaurants and their respective websites that have food under a price rating of 3

SELECT Name, Website

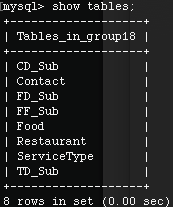
FROM Restaurant

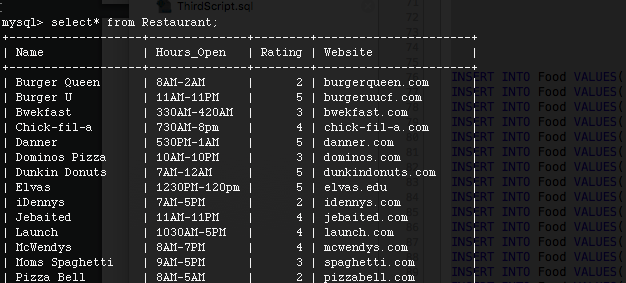
WHERE Name IN (SELECT RestaurantName

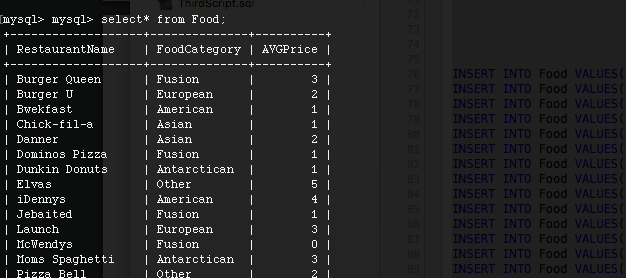
FROM Food

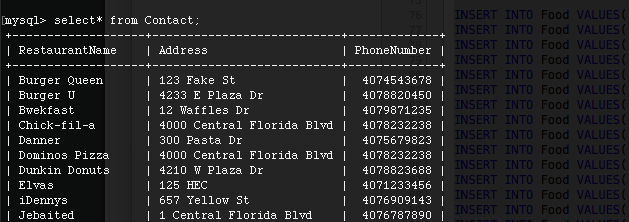
WHERE AVGPrice < 3);

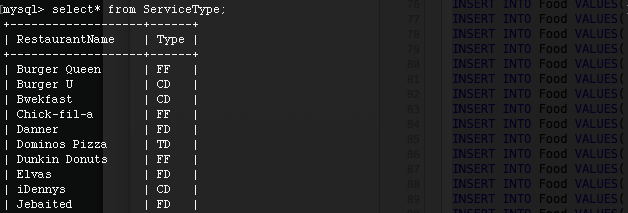
**Populated Relations Screenshots**

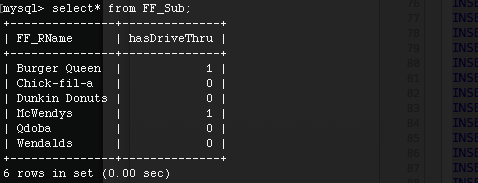


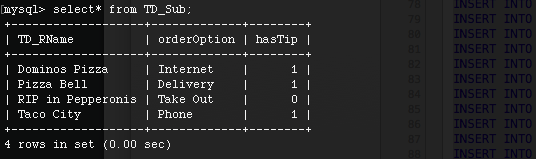
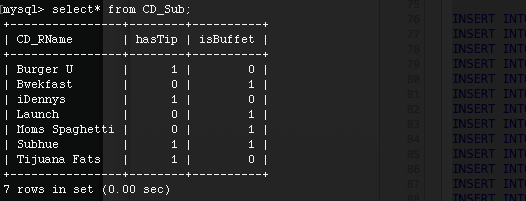
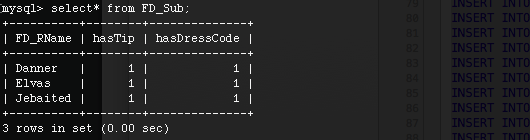






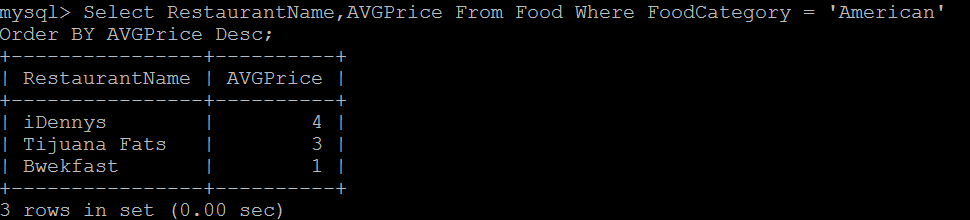


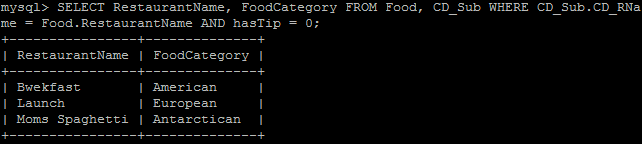


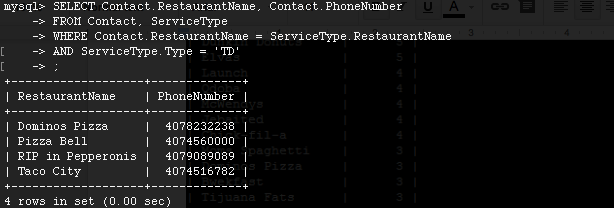
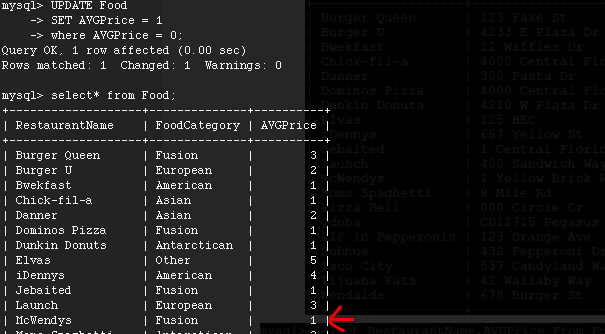
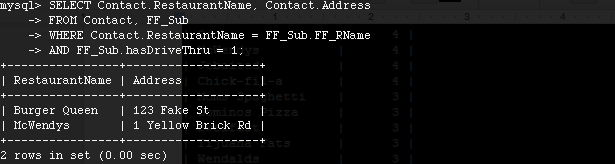
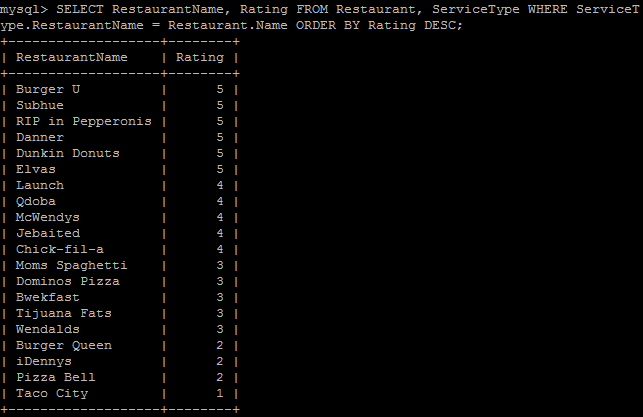


**Database Update Script ScreenShots**

****

****

****

****