**Restaurant(Id, Name, Delivery, Theme):**

Id → Name, Delivery, Theme

Description: This table would allow the user to store all of their restaurants with some general information about them.

FD Analysis: This relation is in 4NF because it satisfies BCNF and there are no multivalued dependencies. Being in BCNF is satisfied when every attribute on the left of a FD is a superkey and in this case that is satisfied as Id is a super key because one Id would only give us one tuple back. None of the attributes on the right are keys as none of them can determine the name of the restaurant with the delivery status or vice versa. Same also goes for themes as restaurants can have the same theme, As well as the same name.

**Employee(employee\_id, restaurant\_id, full\_name, position, weekly\_hours, hourly\_wage):**

employee\_id → full\_name, weekly\_hours, hourly\_wage, resturant\_id

Description: This table would allow the user to store all of their employees across all restaurants, keeping track of their hourly wages and hours worked/planned for that given week. An employee cannot work at two restaurants.

FD Analysis: This relation is in 4NF because it satisfies BCNF and there are no multivalued dependencies. None of the attributes on the right are keys since they cannot uniquely identify any of the other attributes on their own.

**Location(restaurant\_id, country, state\_province, city, street\_address, postal\_code):**

Restaurant\_id → country, state\_province, city, street\_address, postal\_code

Description: This table stores the location of a restaurant.

FD analysis: This relation is in 4NF because it satisfies BCNF and there are no multivalued dependencies. Being in BCNF is satisfied when every attribute on the left of a FD is a superkey and in this case that is satisfied as Restaurant\_id is a super key because one Id would only give us one tuple back. None of the attributes on the right is a key as you can’t determine the other attributes uniquely. For example, the street of the location with the city or vice versa.

**Recipe (Product\_id, Ingredient\_name, unit\_type, quantity)**

Product\_id, Ingredient\_name → unit\_type, quantity

Ingredient → unit\_type, quantity

Description: This table stores the amount of ingredients used in an item along with the unit of that ingredient.

FD analysis: This table is in BCNF because, although there are two functional dependencies, the determiners are both a part of a primary key so BCNF is retained. There are also no Multivalued dependencies. Although at the start we thought there would be, because one product could have multiple ingredients. However we changed that to where to determine a quantity and a unit type of a product for that ingredient and a name for the ingredient was needed.

**Ingredients(ingredient\_name, unit\_cost):**

ingredient\_name → unit\_cost

Description: This relation stores the cost of each ingredient.

FD analysis: Since there are no multivalued dependencies and there are only two attributes this is going to be in 4NF.

**Products(product\_id, product\_name, product\_price, dietary\_resricition\_id, availability):**

product\_id → product\_name, product\_price, dietary\_resricition\_id, availability

Description: This table stores information about an item (product) in the menu such as the price, availability, and etc..

FD analysis: This relation is in 4NF because it satisfies BCNF and there are no multivalued dependencies. Being in BCNF is satisfied when every attribute on the left of a FD is a superkey and in this case that is satisfied as Product\_id is a super key because one Id would only give us one tuple back. None of the attributes on the right is a key as you can’t determine the other attributes uniquely. Product\_name cannot be a key because in this relation an owner can have multiple restaurants that share the same product but with different prices and availability.

**Orders(order\_id, date, time\_placed, time\_expected, status, order\_type, customer\_phone\_number):**

order\_id → date, time\_placed, time\_expected, status, order\_type, customer\_phone\_number.

Description: This table stores information about an order. It would allow the restaurant to better manage the orders as they know when it was ordered, when it is expected and also the order type (take out/delivery/dine-in).

FD analysis: This relation is in 4NF because it satisfies BCNF and there are no multivalued dependencies. Being in BCNF is satisfied when every attribute on the left of a FD is a superkey and in this case that is satisfied as order\_id is a super key because one Id would only give us one tuple back. None of the attributes on the right is a key as you can’t determine the other attributes uniquely.

**Product\_orders(order\_id, product\_id, quantity):**

order\_id, product\_id → quantity

Description: This table allows the user to store the items in an order.

FD analysis: This table is the result of a previously existing multivalue dependency in orders. We originally had an order associated with a single product order in the order table. However, after further analysis this would have resulted in a dependency of order\_id ↠ product\_id. That was not acceptable so we made a many to many relation, which generated this relation where we decided to have both the order\_id and product\_id be the key.

**Order\_types(order\_type):**

order\_type → order\_type

Description: This relation is to make a table for the possible order types which are takeout, delivery, and dine-in.

FD analysis: Since there are no multivalued dependencies and there are only two attributes this is going to be in 4NF.

**Status(status):**

status → status

Description: This relation is to make a table for the possible status types which are ready, delivered, canceled, and received.

FD analysis: Since there are no multivalued dependencies and there are only two attributes this is going to be in 4NF.

**Units(unit\_type):**

unit\_type → unit\_type

Description: This relation is to make a table for the possible unit type.

FD analysis: Since there are no multivalued dependencies and there are only two attributes this is going to be in 4NF.

**Current\_stock(ingredient\_name, restaurant\_id, quantity, unit\_type):**

Ingredient\_name, restaurant\_id → quantity, unit\_type

Description: This table is to keep track of the ingredients left at the restaurants. This would help them better restock their inventory.

FD analysis: This relation is in 4NF because it satisfies BCNF and there are no multivalued dependencies. Being in BCNF is satisfied when every attribute on the left of a FD is a superkey and in this case that is satisfied as ingredient\_name is a super key because one ingredient name would only give us one tuple back as there are no two ingredients with the same name. None of the attributes on the right is a key as you can’t determine the other attributes uniquely. For example, quantity cannot be a key because in this relation two ingredients can have the same quantity.

**Customers(phone\_number, name, address, email)**

phone\_number → name, address, email

email → phone\_number, name, address

Description: This relation is to keep track of their customers. This would allow restaurants to easily access the information of their customers.

FD analysis: This relation is in 4NF because it satisfies BCNF and there are no multivalued dependencies. Being in BCNF is satisfied when every attribute on the left of a FD is a superkey and in this case that is satisfied as phone\_number and email are both super keys because a phone number and an email address would uniquely belong to only one person.

**Dietary\_restrictions(dietary\_restriction\_id, name)**

dietary\_restriction\_id → name

Description: This relation is to make a table for all the possible food dietaries.

FD analysis: Since there are no multivalued dependencies and there are only two attributes this is going to be in 4NF.

**Dietary\_restrictions\_products(dietary\_restriction\_id, product\_id)**

product\_id, dietary\_restriction\_id → product\_id, dietary\_restriction\_id

Description: This is how we relate a dietary restriction with a product.

FD analysis: This relation exists because of the dietary\_restriction and products tables. This must be in 4NF because there are only two attributes, but this table makes it so that both dietary\_restriction and products are in 4NF by removing the multivalue dependencies that existed between them before the existence of this table.