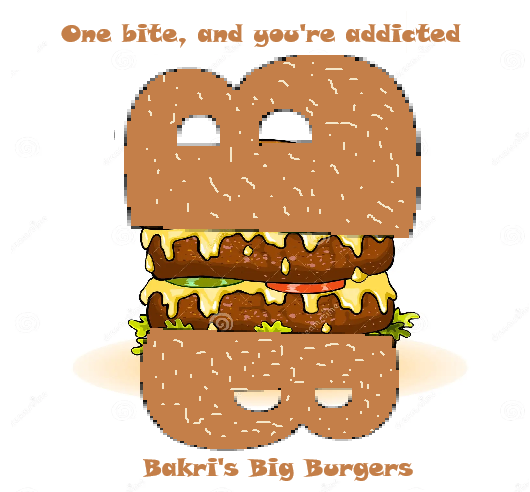
**x**Text

Description automatically generated with low confidence



**A database design for Bakri’s Big Burgers restaurant management system**

By

**Mahmoud Chaer (Team Leader)** (mmc55@mail.aub.edu)

**Hussein Moukalled** (ham69@mail.aub.edu)

**Omar Chehab** (oac05@mail.aub.edu)

A REPORT

submitted to Dr. Hussein Bakri in partial fulfillment of the requirements of phase 2 of the database project for the course EECE433 – Database Systems (Group 6)

October 2024

Table of Contents

[1. Introduction 1](#_Toc180926450)

[2. References/Copyright Section 2](#_Toc180926451)

[3. Tools Used to Draw the ER 2](#_Toc180926452)

[4. System Description & Requirements 2](#_Toc180926453)

[5. Legend of ER Diagram and Symbols 6](#_Toc180926454)

[6. Complete ER Diagram for the BBB Database Management System 7](#_Toc180926455)

[7. New Complete ER 8](#_Toc180926456)

[7.1 Entity Types & their Attributes 9](#_Toc180926457)

[7.1.1 Customer 9](#_Toc180926458)

[7.1.2 Review (weak entity) 10](#_Toc180926459)

[7.1.3 Ingredient 11](#_Toc180926460)

[7.1.4 Supplier 11](#_Toc180926461)

[7.1.5 Meal 12](#_Toc180926462)

[7.1.6 Menu 13](#_Toc180926463)

[7.1.7 Order 14](#_Toc180926464)

[7.1.8 Table 15](#_Toc180926465)

[7.1.9 Chef 16](#_Toc180926466)

[7.1.10 Waiter 17](#_Toc180926467)

[7.1.12 HR 19](#_Toc180926468)

[7.1.13 Administration 20](#_Toc180926469)

[7.1.14 Kitchen\_Station 21](#_Toc180926470)

[7.1.15 Emergency Contact (Weak entity) 21](#_Toc180926471)

[7.2 Relationships and their Explanations 22](#_Toc180926472)

[7.2.1 WRITES (between REVIEW, CUSTOMER) 22](#_Toc180926473)

[7.2.2 PAYS (between CUSTOMER and Order) 22](#_Toc180926474)

[7.2.3 CONTAINS (between ORDER, MEAL) 23](#_Toc180926475)

[7.2.4 DELIVERS (between ORDER, DELIVERY\_DRIVER) 23](#_Toc180926476)

[7.2.5 IS\_MADE\_OF (between INGREDIENT, MEAL) 24](#_Toc180926477)

[7.2.6 IS\_SITTING (between CUSTOMER, TABLE) 24](#_Toc180926478)

[7.2.8 CONTACTS (between WAITER, DELIVERY DRIVER) 25](#_Toc180926479)

[7.2.9 HIRES (between HR, WAITER/DELIVERY\_DRIVER/CHEF) 26](#_Toc180926480)

[7.2.10 COOKS (between CHEF, MEAL) 26](#_Toc180926481)

[7.2.11 MANAGES (between CHEF, KITCHEN\_STATION) 27](#_Toc180926482)

[7.2.12 CONTAINS (between MENU, MEAL) 27](#_Toc180926483)

[7.2.13 SUPERVISION (between CHEF, SOUS-CHEF) 27](#_Toc180926484)

[7.2.14 BOOKS (between TABLE, CUSTOMER, WAITER) 28](#_Toc180926485)

[7.2.15 IS\_ASSIGNED (between TABLE, WAITER) 28](#_Toc180926486)

[7.2.16 DEPENDS\_ON (between EMERGENCY\_CONTACT, CHEF/WAITER/DELIVERY\_DRIVER) 29](#_Toc180926487)

[7.2.17 WORKS\_IN (between CHEF, KITCHEN\_STATION) 29](#_Toc180926488)

[7.2.18 CONTACTS (between ADMINISTRATION, SUPPLIER) 30](#_Toc180926489)

[7.2.19 SUPPLIES (between SUPPLIER, INGREDIENTS) 30](#_Toc180926490)

[7.2.20 SUPERVISION (between SUPERVISOR, SUPERVISEE) 31](#_Toc180926491)

[7.2.21 CREATES (between CHEF, MENU) 31](#_Toc180926492)

[7.2.22 SHIFT (between Administration, chef, waiter) 31](#_Toc180926493)

[8. ER to Relational Mapping 32](#_Toc180926494)

[8.1 Step 1 – Mapping Strong Entity Types 32](#_Toc180926495)

[8.1.1 Relation: HR 32](#_Toc180926496)

[8.1.2 Relation: Administration 33](#_Toc180926497)

[8.1.3 Relation: Menu 33](#_Toc180926498)

[8.1.4 Relation: Meal 34](#_Toc180926499)

[8.1.5 Relation: Supplier 34](#_Toc180926500)

[8.1.6 Relation: Ingredient 35](#_Toc180926501)

[8.1.7 Relation: Chef 35](#_Toc180926502)

[8.1.8 Relation: Delivery\_Driver 36](#_Toc180926503)

[8.1.9 Customer 36](#_Toc180926504)

[8.1.10 Table 37](#_Toc180926505)

[8.1.11 Waiter 37](#_Toc180926506)

[8.1.12 Order 38](#_Toc180926507)

[8.1.13 Kitchen\_Station 38](#_Toc180926508)

[8.2 Step 2 - Mapping of Weaking Entity Types 39](#_Toc180926509)

[8.2.1 Review 39](#_Toc180926510)

[8.2.2 Emergency\_Contact 40](#_Toc180926511)

[8.3 Step 3 – Mapping of Binary 1:1 Relationship Types 40](#_Toc180926512)

[8.3.1 MANAGES 40](#_Toc180926513)

[8.4 Step 4 – Mapping of Binary 1:N Relationship Types 41](#_Toc180926514)

[8.4.1 WORKS IN 41](#_Toc180926515)

[8.4.2 SUPERVISION 41](#_Toc180926516)

[8.4.3 DEPENDS ON 42](#_Toc180926517)

[8.4.4 Contacts 43](#_Toc180926518)

[8.4.5 SUPERVISION 43](#_Toc180926519)

[8.4.6 IS\_SITTING 44](#_Toc180926520)

[8.4.7 IS\_ASSIGNED 44](#_Toc180926521)

[8.4.8 WRITES 45](#_Toc180926522)

[8.4.9 PAYS 45](#_Toc180926523)

[8.4.10 DELIVERS 46](#_Toc180926524)

[8.5 Step 5 – Mapping of Binary M:N Relationship Types 46](#_Toc180926525)

[8.5.1 Contacts 46](#_Toc180926526)

[8.5.2 Contains 47](#_Toc180926527)

[8.5.3 Cooks 47](#_Toc180926528)

[8.5.4 Composed\_Of 47](#_Toc180926529)

[8.5.5 Creates 48](#_Toc180926530)

[8.5.6 Hires 48](#_Toc180926531)

[8.5.7 Supplies 48](#_Toc180926532)

[8.6 Mapping of Multivalued Attributes 49](#_Toc180926533)

[8.6.1 Multivalued Attribute: Assigned\_Location 49](#_Toc180926534)

[8.6.2 Payment Method 49](#_Toc180926535)

[8.6.3 Image 50](#_Toc180926536)

[8.6.4 Image 50](#_Toc180926537)

[8.6.5 Product 51](#_Toc180926538)

[8.6.6 Day 51](#_Toc180926539)

[8.7 Mapping of N-Ary Attributes 52](#_Toc180926540)

[8.7.1 Books 52](#_Toc180926541)

[8.7.2 Shift 52](#_Toc180926542)

[8.7.3 Places 53](#_Toc180926543)

[8.8 Step 8- Mapping Aggregation, Specialization Relationships 53](#_Toc180926544)

[9. Final Display - All Tables 54](#_Toc180926545)

[10. Tables’ States 55](#_Toc180926546)

[10.1 HR 55](#_Toc180926547)

[10.2 Administration 56](#_Toc180926548)

[10.3 Menu 57](#_Toc180926549)

[10.4 Meal 58](#_Toc180926550)

[10.5 Supplier 59](#_Toc180926551)

[10.6 Ingredient 59](#_Toc180926552)

[10.7 Chef 60](#_Toc180926553)

[10.8 Delivery\_Driver 60](#_Toc180926554)

[10.9 Customer 61](#_Toc180926555)

[10.10 Table 62](#_Toc180926556)

[10.11 Waiter 62](#_Toc180926557)

[10.12 Order 63](#_Toc180926558)

[10.13 Kitchen\_Station 64](#_Toc180926559)

[10.14 Review 64](#_Toc180926560)

[10.15 Emergency\_Contact 65](#_Toc180926561)

[10.16 Payment\_Method 65](#_Toc180926562)

[10.17 Image\_Meal 66](#_Toc180926563)

[10.18 Image\_Review 66](#_Toc180926564)

[10.19 Product 67](#_Toc180926565)

[10.20 Day 67](#_Toc180926566)

[10.21 Assigned\_Location 68](#_Toc180926567)

[10.22 Supplies 68](#_Toc180926568)

[10.23 Creates 69](#_Toc180926569)

[10.24 Composed\_Of 69](#_Toc180926570)

[10.25 Cooks 70](#_Toc180926571)

[10.26 Contacts 70](#_Toc180926572)

[10.27 Places 71](#_Toc180926573)

[10.28 Hires 71](#_Toc180926574)

[10.29 Shift 72](#_Toc180926575)

[10.30 Books 72](#_Toc180926576)

[10.31 Contains 73](#_Toc180926577)

[11. Conclusion 73](#_Toc180926578)

[12. Instructor’s Feedback 73](#_Toc180926579)

# Introduction

A restaurant management system database is a crucial element of such a company that could very well be the difference between branching and bankruptcy. The importance of such a database system from the need for efficient and streamlined operations in the fast-paced restaurant environment. The proper functioning of a restaurant relies on a complex interconnection of operations such as order management, inventory tracking, employee management, and customer service which are all time consuming and error prone tasks. A robust database system addresses these challenges by providing a centralized platform that organizes and manages all the critical components mentioned above ensuring seamless communication and organization between all different departments, roles and components of the company. By opting for a performant, strongly built database, the business avoids frequent complications such as overbooking, stock shortages and miscommunication to list a few.

Such a database would represent the restaurant itself as a mini world; its day-to-day activities including customers, employees’ menus and administrative tasks. It would be of great importance to the owner(s) of the restaurant as it would allow them easy access to any information of value such as reviews, employees, stocks etc… making it easier to evaluate performance and make decisions to ensure maximum profitability. It would also assist the Human Resources department in making informed decisions about who to promote, demote or fire based on performance reviews and readily available stats. As for the staff, they require the system to efficiently track customer orders, follow table assignments and bookings, access data of interest such as recipes, review customer preferences and check customer feedback. Finally, customers' expectations from such a system would be smooth order deliveries, prompt service through seeming less communication between the kitchen staff and the waiters, easy reservation systems and an overall excellent dining experience.

Specific requirements for the database differ between stakeholders. Owners and managers would expect quick and easy access to key performance indicators such as customer reviews, shift assignments, number of orders and stock levels. Staff such as waiters and chefs would expect this system to facilitate quick and precise order management, table reservations and order deliveries to always ensure efficiency. As for customers, an inviting user interface that makes it easy to check menus, book tables, order food and leave reviews is essential. By addressing all these diverse needs, a database system would promote operational efficiency and consequently drive the restaurant’s profitability.

This report describes the Bakri’s Big Burgers restaurant main database design using the Entity-Relationship diagram approach. Section 2 lists the sources and references used in the making of this project, section 3 shows the tool(s) used to draw the ER, section 4 describes the system and it’s requirements, section 5 is a legend of the ER diagram symbols, section 6 describes the complete ER diagram by explaining all entities with their respective attributes (6.1) and all relationships with their attributes (6.2), finally section 7 is the conclusion.

# References/Copyright Section

* Elmasri, R., & Navathe, S. B. (2016). Fundamentals of database systems (7th ed.). Pearson.
* Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). Database system concepts (7th ed.). McGraw-Hill.
* Ramakrishnan, R., & Gehrke, J. (2002). Database management systems (3rd ed.). McGraw-Hill.
* Boardmix. (n.d.). ER diagrams: The complete guide to database modeling.

# Tools Used to Draw the ER

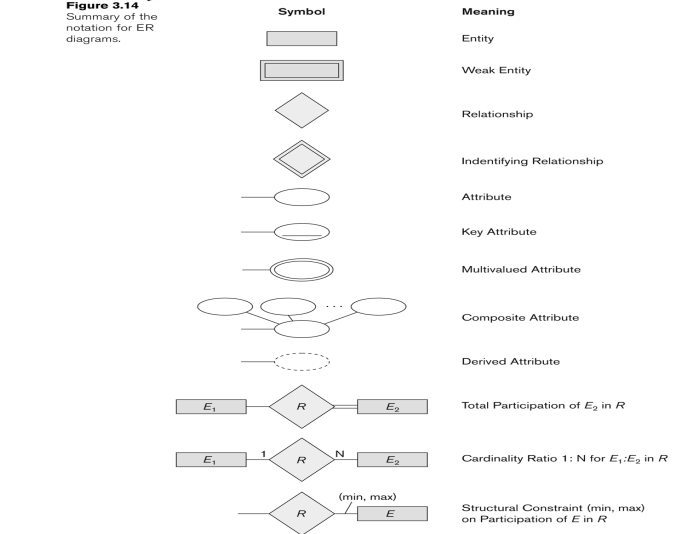
For this phase of the project, we only used the software draw.io (<https://draw.io>) to draw our ER.

# System Description & Requirements

The proposed restaurant database design promotes easy and coordinated data storage and tracking to facilitate the overall operations of the business. The following are the system requirements:

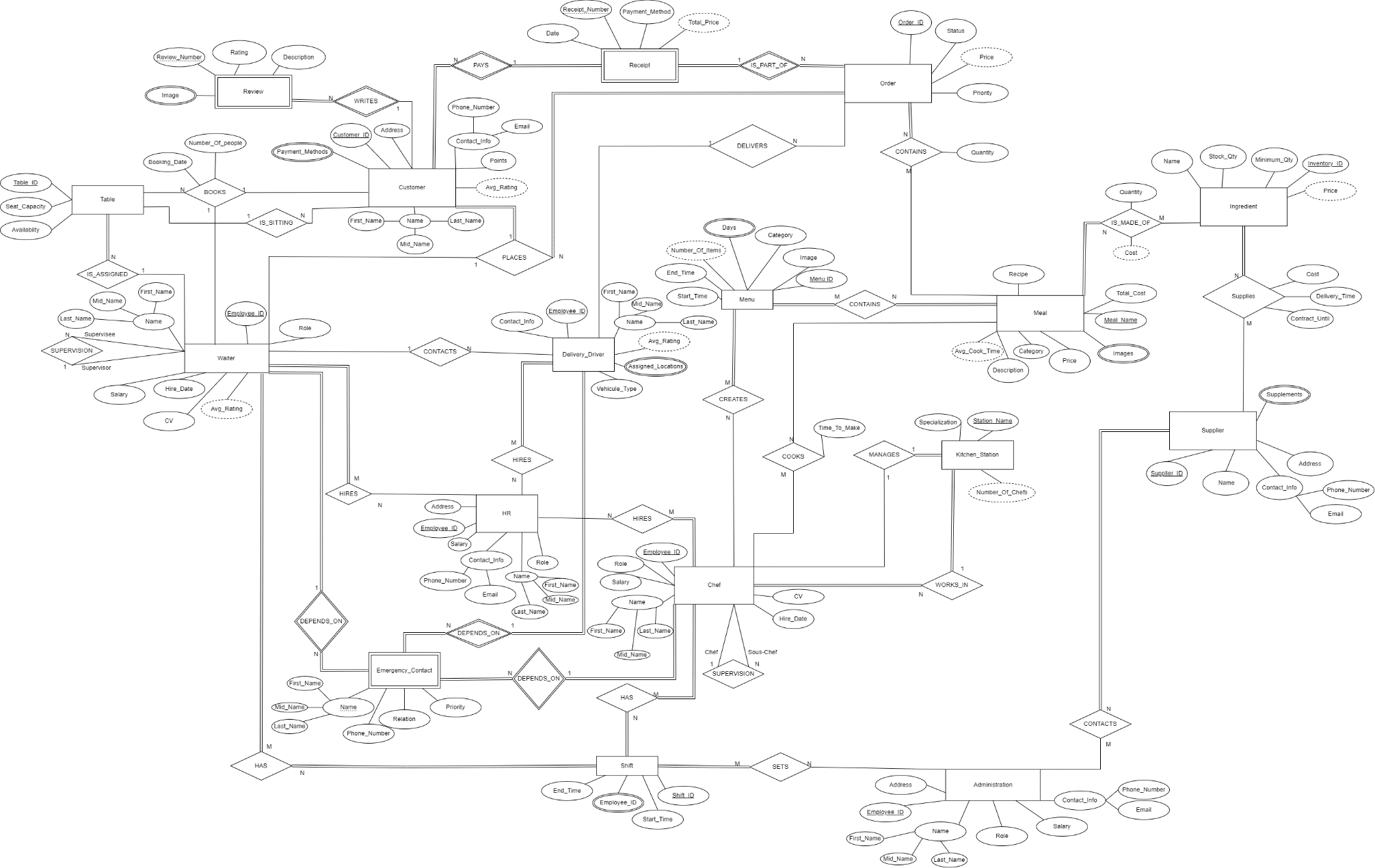
* Each staff member is registered in the database as a waiter along with all of its attributes. To differentiate between staff, we look at the attribute role of the waiter. The Avg\_Rating attribute is derived from the avg rating that this specific waiter got during the reviews as reviews are made to specific waiters. The Hire\_Date is the date when the waiter was hired. The salary is the waiter’s current salary. The CV is the current CV of the waiter. The Employee\_ID is just a primary key assigned to each waiter in order to identify them.
* All the tables of the restaurant are registered in the database. The Availability is whether the table is empty. Seat\_Capacity is how many people can sit on this Table. The Table\_ID is just an identifier for the table. Each waiter is assigned specific tables that he must take care of.
* Every Order that is placed by a Customer and Waiter in the restaurant is stored in the database. To identify each unique Order, we use the Order\_ID which is the primary key. Status is used to know whether the Order is pending, in progress or done. The priority is to rank the orders in terms of importance, this helps the chefs in knowing what orders to start preparing first. Finally, we have the derived attribute Price, which indicates the price of a certain Order, it is derived from the Meal entity. At the end, this Order in paid by the Customer that placed the Order, using a payment method of his choice.
* Every Meal that is cooked by a Chef in the restaurant is stored in the database. To identify each unique Meal, we use a Meal name, which serves as the primary key. The Recipe indicates the ingredients used to make the Meal along with the amount used. We have two similar attributes that are Price and Total\_Cost, Price represents the amount of money that the Customer has to pay in order to get this Meal, while Total\_Cost represents the amount of money used to make this Meal. The Image attribute represents a visual representation of every Meal, it is multivalued since we can have several photos for a single Meal. Category represents the type of the Meal, whether it is an appetizer, a salad, a main dish… The description attribute describes the Meal and how it is served in a sentence. Finally, the derived attribute Avg\_Cook\_Time represents the average time for a Meal to be cooked, it is derived from the relationship COOKS between chef and Meal.
* Every Customer that Places an order is stored in the database. In order to identify every unique Customer, a Customer\_ID is used, which is the primary key. The customer’s Name is stored in the database. Address simply represents the location of the Customer’s home. The contact\_Info of the Customer is also stored in the database, it represents the Customer’s phone number and email address, it is a composite attribute. The Payment\_Methods usually used by the Customer are stored in the database, it is a multivalued attribute that can include cash or/and cards. A Customer can gain Points by coming to the restaurant frequently, these points are stored in the database and can get you a discount when they reach a certain amount. Finally, the Avg\_Rating, it is a derived attribute which represents the average rating that a Customer leaves after every visit, it is derived from the Review entity.
* Every Chef employed at the restaurant is stored in the database which can be identified by an Employee\_ID. His Name is stored in the database with many other attributes. The CV represents the Chef’s past experiences and his level of education. The date on which the Chef was hired is stored in the database as Hire\_Date. The Chef’s Salary is stored in the database, which represents the amount of money that he gets by working at the restaurant. Finally, the Role attributes represent the occupation of the Chef inside the kitchen (baker, fry chef…).
* Every Delivery\_Driver that works for the restaurant is stored in the database which can be identified by an Employee\_ID. The Driver’s Name is stored in the database. The contact\_Info of the driver is stored in the database, it represents the Driver’s phone number. The regions/locations that are covered by every driver are stored in the database as assigned\_locations, it is a multivalued attribute since a driver can cover several locations. Vehicule\_Type represents the vehicle used by the driver whether it is a motorcycle, a sedan or a truck for large deliveries. Finally, the Avg\_Rating, it is a derived attribute which represents the average rating that customers leave for the driver, it is derived from the Review entity.
* Every Ingredient that is Supplied to the restaurant by a Supplier is stored in the database along with an Inventory\_ID, which helps in identifying every unique ingredient. The Stock\_Qty represents the available quantity of a certain Ingredient in stock, while Minimum\_Qty represents the minimum quantity that should be available for a certain Ingredient in stock. if the Stock\_Qty is less than Minimun\_Qty, Ingredients should be supplied by the Supplier. The Price of Ingredients is stored as a derived attribute, which simply represents the Price of a certain Ingredient, it is derived from the attribute Cost in the SUPPLIES relationship between Ingredients and Supplier. The Name of Ingredients is stored in the database as well.
* Every Supplier that worked with the restaurant is stored in the database along with a Supplier\_ID used to identify him. The Supplier’s Name is stored in the database, it represents the company’s Name. The contact\_Info of the Supplier is stored in the database, it represents the Supplier’s phone number and email address. Address simply represents the location of the Supplier’s factory. The Products that are supplied by him are stored in the database as a multivalued attribute since a Supplier can supply more than one product.
* Every Kitchen\_Station that has chefs working in it is stored in the database along with its name Station\_Name to identify each station. The number of chefs that are working in a certain station is stored in the database and represented as Number\_Of\_Chefs (derived attribute as its data is already stored in the database from the relationship works, between Chef and Kitchen\_Station). Finally, the Specialization of every station is stored in the database, a station can be specialized in desserts, seafood, and so on.
* Every Review that was written by a customer is stored in the database along with a Review\_Number to identify each Review. The presence of a Review depends on the presence of a Customer since a Review cannot be written without a Customer. The Rating that the Customer left reflects his overall experience in the restaurant, this is usually followed by a Description where the Customer describes his experience in a couple of sentences. Both of these attributes are stored in the database. Finally, the multivalued attribute Images represents the photos that the Customer took during his stay at the restaurant.
* An administration is responsible for overseeing the overall operations in the restaurant and each administrator is identified by a unique Employee\_ID. The name of an administrator is stored in the database along with his contact info which is composed of an email and a phone number for easy and quick reachability at any time. We also keep track of the address of the administrators for live correspondence. Finally, each administrator is assigned different roles and therefore earns different salaries. The Administration also sets a Shift for every employee, in which the start and the end time of the Shift are included.
* The HR department oversees the recruitment and hiring process of every employee in the restaurant. Each HR employee has a unique Employee\_ID that helps identify it as well as a name. HR members also have their contact info stored as a phone number and email as well as their address for easier correspondence. Roles vary between HR employees and so do their salaries.
* A menu contains the served meals at any given time, each menu is identified by a unique Menu\_ID. Different menus are given at different times of the day i.e. breakfast, lunch, dinner there for each menu has a start and end time as well as a category section, and image and the Number\_of\_items in it. Finally, the Days during which a particular menu is served is recorded to know when to serve what menu.
* Every Emergency\_Contact that is linked to an employee (Chef/Waiter/Delivery\_Driver) is stored in the database along with the Name of the contact which is used for identification purposes. The attribute Relation simply represents the relation between the employee and his Emergency\_Contact, such as friends or family. The contact’s Phone\_number is stored in the database, it allows the restaurant to reach the Emergency\_Contact. Finally, the priority attribute ranks the Emergecy\_Contacts of a certain employee from most to least important.

# Legend of ER Diagram and Symbols

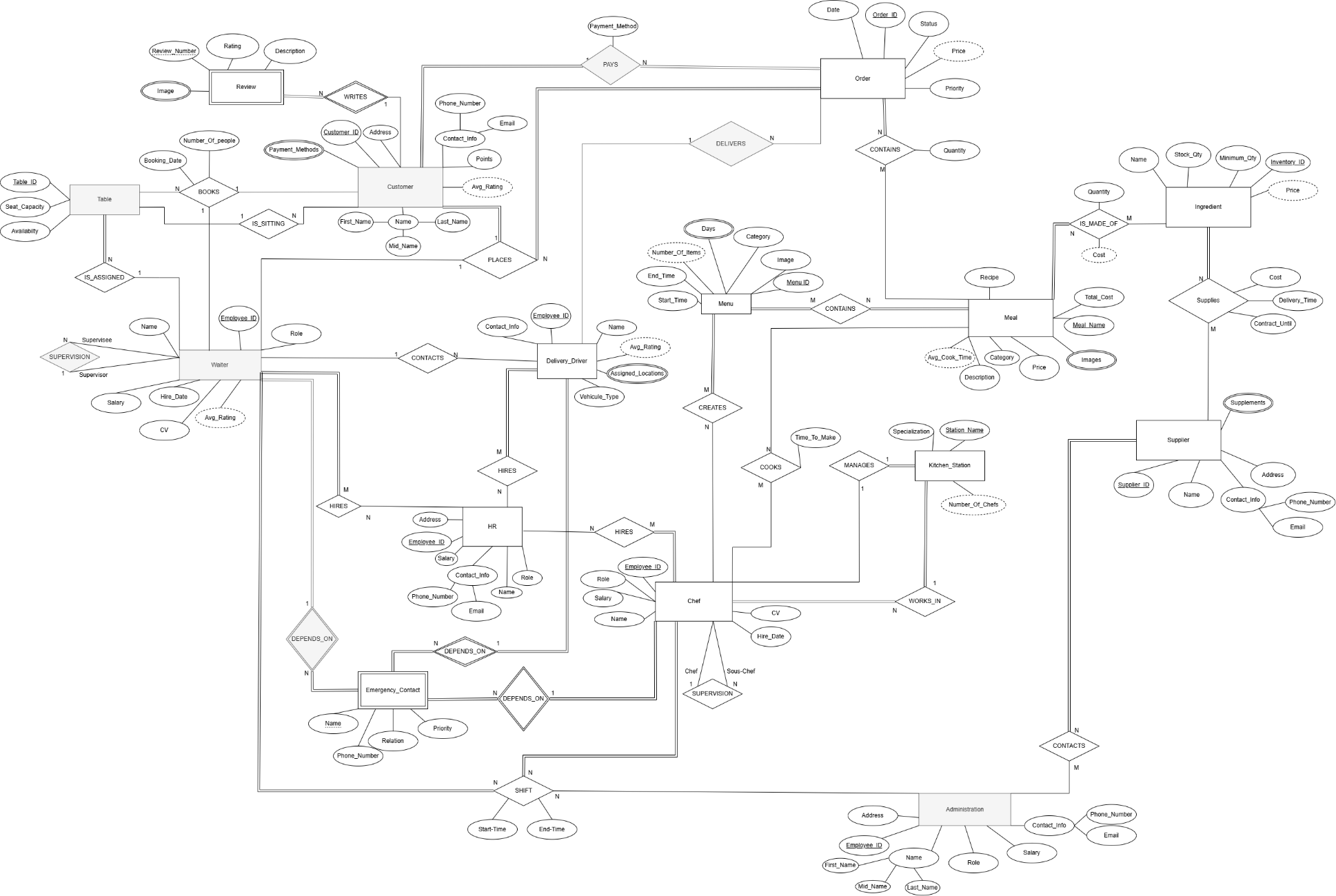


Source : Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of database systems* (7th ed.). Pearson

# Complete ER Diagram for the BBB Database Management System



# New Complete ER



## Entity Types & their Attributes

### Customer

The Customer is any person in the restaurant that is ordering Meals. A restaurant must have important information about its customers to know how to work with them in the future. This information are saved in the attributes:

* Customer\_ID (primary key):

It is the primary key of the Customer entity. It is used to target a specific Customer in case of a database query.

* Name:

The attribute Name is needed for communication and differentiating between the different employees and customers. Also used to print the name tags of the employees.

* Contact\_Info (composite Phone\_Number, Email):

This info is needed so that the restaurant can contact each customer whenever they wish to. It might be used in the event of a reservation or a special offer, or just generally for advertising. Email is more for formal messages. Phone\_Number is for faster responses (in case the Customer is late for a reservation for example).

* Points:

The Points are saved for each Customer. This is a reward system were based on the amount of Points a Customer has, the restaurant can know if they should give them a discount or special treatment (high points shows that the Customer comes a lot to the restaurant).

* Address:

The Address of a Customer is needed in case future deliveries might occur or in case the restaurant wants to send a gift or reward for the Customer.

* Payment\_Methods (multivalued attribute):

This info contains the payment methods that the Customer has used previously. The attribute is multivalued because a Customer might have used several payment methods in the past (ex: Credit Card, Cash, Crypto, ect…). This attribute is needed because for example, if a Customer places a delivery order, the restaurant might need to know if the delivery person must have additional cash on him or for credit card payments, the credit card device.

* Avg\_Rating (derived attribute):

Avg\_Rating is the average review rating that the customer has left on all his reviews. It is a derived attribute since there is no need to enter it again as it can be computed from the average rating from the reviews. It is needed by restaurants to make sure that all the customers are always satisfied. In case a Customer has a low Avg\_Rating, the restaurant can make sure that whatever caused this bad rating doesn’t happen again. Maybe also by giving him gifts or discounts to compensate for the low rating.

### Review (weak entity)

The Review entity is used for storing the reviews that the customers leave. Each restaurant wants to be able to view the different reviews they got to improve the areas where they are performing poorly or not enough.

* Review\_Number (partial key):

This is used as a key (along with the customer that left it) in order to be stored in the database.

* Rating:

This is the rating left by the Customer, the rating is over 5 (/5). Each review should have a rating to be able to visualize a numerical value that reflects the customer’s feedback.

* Image (multivalued attribute):

This attribute is needed so that the Customer leaves images of his review (showing proof). The images must be stored for future analysis. It needs to be multivalued, as a user might leave several images in one Review.

* Description:

This is the text describing the Review that the Customer left. It contains the Customer’s feedback, message, or any other text that the Customer might want to leave. It is important as images and ratings are most of the time not enough to show the Customer's true feedback.

### Ingredient

This entity is for the ingredients that are used to produce meals. Each ingredient has its own info that is saved in the database.

* Inventory\_ID (primary key):

This is just an ID that is used to track the ingredients in the inventory.

* Name:

Name of the Ingredient. It is not the primary key, as some ingredients might have the same name but are different.

* Stock\_Qty:

Used to track the stock quantity left of a specific ingredient in the inventory.

* Minimun\_Qty:

This attribute is needed so that the restaurant knows what the minimum quantity is needed in stock of a specific ingredient so that they can order a restock when they reach this quantity.

* Price (derived attribute):

It is a derived attribute as it can be directly taken from the other attribute (Cost in the Supplies (Ingredient/Supplier) relationship). It reflects the cost of this ingredient. This attribute might be needed here so that whenever a user needs to get the cost of an Ingredient, he can directly get it from the entity. In this way, the ingredient entity has all the info needed about the ingredients.

### Supplier

The Supplier entity contains the info related to a supplier that the restaurant gets items from.

* Supplier\_ID (primary key):

This is just the primary key of the Supplier as other attributes are not always unique.

* Name:

This attribute is the name of the company that acts as a supplier. It is needed so that the restaurant can contact them.

* Contact\_Info (composite Phone\_Number, Email):

This attribute contains the contact info of the Supplier. This info is needed so that the restaurant can contact the Supplier in case they need a restock or any other emergency demand. Sometimes Email is not enough, for faster response Phone\_Number might be needed.

* Address:

Address of the Supplier. Needed in case the restaurant wants to send an employee to pick up some items or in case a meeting is happening.

* Products (multivalued):

This attribute stores the item that the Supplier supplies to the restaurant. It is a multivalued attribute because a Supplier might supply multiple items.

### Meal

The Meal entity is storing info related to one meal. For example, a Meal can be a pizza, burger, ect…

* Meal\_Name (primary key):

The Meal\_Name is used as a primary key to distinguish between the other meals.

* Recipe:

The actual recipe needed to prepare a Meal. A recipe is needed so that the chefs can know how to prepare a specific Meal.

* Price:

It is the price that the Meal is selling for in the restaurant. The price is needed so that the customers know how much each Meal is going for.

* Category:

The Category of a Meal is its type (ex: side, main course, drink…). It is important as it helps in dividing the meals into different categories so that they can be distinguished.

* Description:

Describes what the Meal is about. It can contain what ingredients are used, how it is prepared and what to expect from it. This describes to the Customer the Meal that he will be ordering.

* Images (multivalued attribute):

These are the images that are stored for a Meal. A Meal can have several Images, this is why this attribute is multivalued. The images need to be stored so that they can be displayed to the Customer for a better vision of the Meal.

* Total\_Cost:

This attribute is the total cost that is needed to prepare the meal. This attribute cannot be derived since there are several hidden costs such as power costs that we don’t have in other entities.

* Avg\_Cook\_Time (derived attribute):

This attribute is the average time needed to prepare a Meal. It is a derived attribute because we can get it from averaging the Time\_To\_Make (time taken for a meal to be prepared by a Chef).

### Menu

This is the set of meals that are given in a specific period. For example, during breakfast time, a different menu might be used compared to during lunch time.

* Menu\_ID (primary key):

Primary key, used to identify menus. The Menu Category might not be enough because, for example, we might have a breakfast Menu Monday different from the breakfast Menu Sunday.

* Image:

An image that is the cover of the Menu. It can be seen as a main image for the Menu.

* Category:

This attribute is the type of Menu. Some examples are breakfast and lunch. This is important to distinguish between different types of menus.

* End\_Time:

The time when the Menu is no longer available.

* Start\_Time:

The time the Menu starts being available at.

* Days (multivalued):

The days for which the Menu is being used. It is a multivalued attribute because a Menu can be used for several days.

* Number\_Of\_Items (derived attribute):

The number of meals in a specific Menu. It can be needed to quickly see if a Menu needs more meals in it or less. The value is derived as we can directly compute the number of meals that share the relationship Contains.

### Order

This entity represents the details of a Customer’s request for food, each time the Customer requests something, an Order is made. At the end of each sitting, all the Customer’s orders are added in one Receipt for him to pay. It contains several attributes stored in the database such as:

* Order\_ID:

This attribute is the Primary Key of the Order entity, it is used as an identifier for each unique order in the database.

* Status:

This attribute tracks the current state of the order, it helps the restaurant in tracking each order and knows whether it is facing a delay or not, which could help in identifying a problem and improving efficiency. This is a great addition to keeping the Customer updated on the Status of his Order.

* Priority:

This attribute ranks Orders that are placed at around the same time by priority. This could help the restaurant in keeping every Customer satisfied by not having to wait.

* Price (derived):

This attribute represents the sum of every Meal in our Order, it helps the Restaurant in getting the total price in the Receipt. It is derived from the Meal entity by adding the price of every Meal in the Order.

### Table

This entity represents seating arrangements in the restaurant, it helps the restaurant in managing reservations for Customers and tracking Orders associated with each Table. It contains several attributes stored in the database such as:

* Table\_ID:

This attribute is the primary key for the Table entity, it helps the restaurant in identifying each unique Table.

* Seat\_Capacity:

This attribute represents the number of people that can be seated on a table, this helps the restaurant in the reservation process.

* Availability:

This attribute represents the status of the table, whether it is occupied, available or reserved. This helps the restaurant in organizing everything related to reservations.

### Chef

This entity represents the individual that is responsible for preparing meals, creating menus and managing kitchens. A restaurant cannot run properly without chefs. It contains several attributes stored in its database such as:

* Employee\_ID:

This attribute represents the primary key of the Chef entity, it helps in identifying each unique chef.

* Name:

Each chef has a name which is shown on the tag pinned on his cloth.

* Role:

This attribute represents the different roles that a chef can have such as baker, fry chef and so on. This helps the restaurant to know how many chefs of each kind are available in the restaurant to know whether we have a shortage or not.

* Salary:

Each Chef is paid a salary at the end of the month, the salary varies between a Chef and another depending on his Role. This attribute helps the restaurant in keeping track of their financial records.

* CV:

Each Chef submits his CV which is thoroughly studied by the HR to hire only the finest applicants.

* Hire\_Date:

The restaurant keeps track of the Hiring date of each Chef, it makes it easier to promote him later or give him a raise based on his years of service for the restaurant.

### Waiter

This entity represents a staff member in the restaurant who interacts with customers, takes orders and books tables. No restaurant is complete without the smooth dining experience offered by a competent waiter staff.

* Employee ID (Primary key):

Primary key, used to identify the waiter; each waiter has a uniquely assigned Employee ID to help better identify him when he’s hired.

* Role:

Each waiter is assigned a different role which makes it easier to identify the speciality and job of each waiter; some waiters are bar waiters and fix drinks for the customers, others are responsible for taking orders and serving the food and some are responsible for taking calls and booking tables.

* Name:

Each waiter has a name which shows on the tag pinned to his cloth.

* Salary:

Each waiter is paid a salary at the end of the month, the salary varies between a waiter and another.

* Hire\_Date:

We keep track of the Hiring date of each waiter, it makes it easier to promote him later or give him a raise based on his years of service for our restaurant.

* CV:

Each waiter submits his CV, which is thoroughly studied by the HR to hire only the finest applicants.

* Avg\_Rating (Derived):

Each waiter has an average rating based on comments and reviews customers leave, this helps ensure that the restaurant’s employees are meeting the expected standard and acts as an extra factor in the decision to promote or demote a waiter.

* + 1. Delivery Driver:

This entity represents an employee who delivers food orders from the restaurant to the customer. In case the customer lives far away from the restaurant or just feels like staying at home, he can have his food delivered to his doorstep with timely and accurate order fulfillments.

* Employee ID (Primary Key):

Primary key used to identify the waiter; each delivery driver has a uniquely assigned Employee ID when he’s hired to help better identify him.

* Contact\_Info:

A delivery driver provides his phone number so that the restaurant and the customers can contact him, making the overall experience smoother.

* Vehicle\_Type:

The vehicle used varies between a driver and another, for far away distances the order would most likely be assigned to a driver who uses a car while shorter distance orders will be taken by drivers on motorcycles or bikes.

* Salary:

Each delivery driver is paid a salary at the end of the month, the salary varies between a delivery driver and another depending on how many areas he covers and his rating.

* Name:

Each delivery driver has a name which is given to the customer when he places an order so that he can familiarize himself with the person responsible for the delivery of his order and allow both the waiter and the customer to contact him.

* Avg\_Rating (Derived):

Each delivery driver has an average rating based on comments and reviews customers leave, this makes it easier to maintain the standard required by the employees and plays a factor in the decision of promoting or demoting said employee.

* Assigned\_Locations (Multivalued):

The covered locations by the restaurant are divided on the delivery drivers to make the process more efficient and ensure timely delivery of the orders, each driver can be responsible for more than 1 area.

### HR

The HR (Human resources) is the department responsible for managing the employee-related aspect of the restaurant, its main role is to interview and hire the most competent participants to ensure an excellent restaurant staff.

* Employee ID (Primary Key):

Primary key used to identify the HR employee; each HR employee has a uniquely assigned Employee ID to help better identify him.

* Name:

Each HR employee has a name divided into First name, Middle name and Last name to help identify and contact him.

* Address:

The address of each member of the HR department could come in handy for official correspondence through documents like letters or tax forms.

* Salary:

Each HR employee is paid a salary at the end of the month, the salary varies between an HR employee, and another based on his role

* Contact info (Composite):

Each HR member has contact info which comes in the form of his email and his phone number so that employees can contact him for referrals, ideas or complaints.

* Role:

Each HR member has a different role, some could be responsible for the hiring decisions and the recruitment, others could be responsible for handling complaints. Giving a role to each HR employee makes the department more structured and efficient.

### Administration

The administration is a crucial department responsible for overseeing the overall operations and management of the restaurant. It takes care of the higher business details and ensures the smooth functioning of all the fractions of the restaurant.

* Employee ID (Primary Key):

Primary key used to identify the administrators; each member of the administration has a uniquely assigned Employee ID to help better identify him.

* Name:

Each member of the administration has a name to help identify and contact him.

* Address:

The address of each member of the administration is given, which could come in handy for official correspondence through documents like letters or tax forms.

* Role:

Each administrator has a different role, some could be responsible for overseeing the operations of the HR department, some could be responsible for the business decisions ensuring a smooth functioning of the administration.

* Salary:

Each member of the administration is paid a salary at the end of the month, the salary varies between an administrator, and another based on his role

* Contact\_Info:

Each member of the administration has contact info which comes in the form of his email and his phone number to facilitate access to him and make it easier to be in contact with brands, make deals or talk to others.

### Kitchen\_Station

This entity is used to monitor the division of the kitchen between chefs and different tasks. Each kitchen station has a different purpose and is assigned one or more chefs. This helps the restaurant divide the work and follow up with any issue by targeting the specific station where the issue came from.

* Station\_Name (primary key):

This is the name of the station which can also help identify the Kitchen\_Station in the database.

* Number\_Of\_Chefs (derived attribute):

This attribute shows how many chefs work in a targeted Kitchen\_Station. This can be beneficial the restaurant to keep track of the number of employees in each station compared to the amount of work that it produces.

* Specialization:

This attribute shows the restaurant what each Kitchen\_Station specifically does (ex: meat station, dough preparation, ect…). This can help the staff quickly send the tasks to the needed station.

### Emergency Contact (Weak entity)

An emergency contact is crucial for every employee in any company, this person will be contacted in case of an emergency. Many emergency contacts should be given with an order of priority to ensure the highest chance of at least one of them answering.

* Name (Partial Key):

The name of the emergency contact is provided and divided in First name, Middle name and Last name to better identify him and make contacting him easier, it is stored as partial key because it might not be unique.

* Phone\_Number:

Each emergency contact has a phone number so that he can be contacted as quickly and easily as possible in case of an emergency.

* Relation:

The relation of the emergency contact to the employee, is he a family member, a friend, a coworker… to make the interaction smoother.

* Priority:

The priority of an emergency contact describes in what order should he be contacted; a higher priority means this emergency contact will be contacted first.

## Relationships and their Explanations

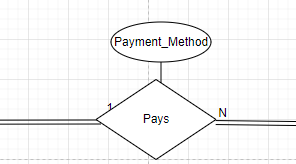
### WRITES (between REVIEW, CUSTOMER)



Total participation from Review, partial participation from Customer. This is a binary relationship where each Customer can write one or more reviews. But a Review must be written by a Customer. A Review can only be written by one Customer. This is how reviews work in general. A user can write several reviews, and each review has only one author.

This relationship helps the restaurant know which Customer has written a Review in order to help them fix a bad Review.

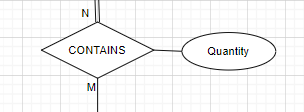
### PAYS (between CUSTOMER and Order)



Total participation from both Customer and Order. This is a binary relationship where a Customer must pay for one or more orders and an order must be paid for by a customer. An order must be paid by only one Customer, if several customers paid for an order only one Customer is saved as the one who paid.

The Payment\_Method attribute is used to save which payment technique the customer uses, i.e. cash, card, vouchers.

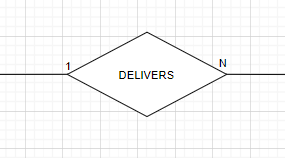
### CONTAINS (between ORDER, MEAL)



Total participation from Order, partial participation from Meal. This is a binary relationship where each Order needs to contain one or more meals. But it is not necessary that all meals are part of an Order. A Meal can be part of no Order. A Meal can be part of many orders. Each relationship between these two is characterized by a quantity. Because an Order has many meals, but the quantity shows the restaurant the quantity of a specific Order.

This relationship is important as it helps the restaurant know each Order is made of what meals to directly give it to the chefs to prepare it.

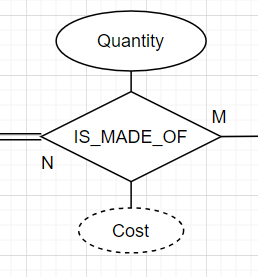
### DELIVERS (between ORDER, DELIVERY\_DRIVER)



Partial participation from both Order and Delivery\_Driver. This is a binary relation between Delivery driver and order where a Delivery\_Driver can deliver one or more orders. And an Order is delivered by only one driver. An Order is not necessarily delivered since the Order might be for the restaurant locally not to an address outside. And the Delivery\_Driver doesn’t necessarily deliver an Order all the time.

This relationship is important as it helps the restaurant know which Order each Delivery\_Driver has delivered and follow up with them in case of an issue.

### IS\_MADE\_OF (between INGREDIENT, MEAL)



Total participation from Meal, partial participation from ingredients. This is a binary relationship where each Meal is made of one or more ingredients. Each Ingredient is part of one or more meals. A Meal must be made of ingredients, but not all ingredients are necessarily part of a Meal (maybe some ingredients are only for the staff to eat from). Each relationship between Meal and Ingredient is characterized by a Quantity and Cost. Because a Meal can have different quantities of different ingredients. And a Cost can be useful to directly get the cost of the ingredients of this Meal. It is a derived attribute since we can directly get it by summing the costs of the ingredients used.

This relationship is important as we can see directly what each Meal is made of, as well as its ingredients’ cost.

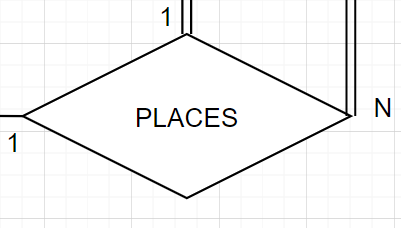
### IS\_SITTING (between CUSTOMER, TABLE)



Partial participation from Customer, parietal participation from Table. This is a binary relationship where one customer sits on one table and one table can have many customers sitting on it. Not all customers sit on a table, they could have their food to go or order it. Not all tables need to have customers sitting on them, some could be empty.

This relationship shows that the customers have a dine in option where they can sit on a table and eat their food in the restaurant.

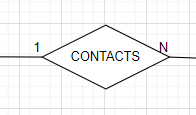
* + 1. PLACES (between WAITER, CUSTOMER, ORDER)



Total participation Customer and Order, partial participation from Waiter. This is a ternary relationship where one customer contacting one waiter can place many orders, one waiter takes one order from one customer and one customer placing one order talks to one waiter. All customers must place an order, and all orders must be placed, but not all waiters take orders.

This relationship shows that to get the food, whether for dine in or delivery, a customer has to order it via a waiter.

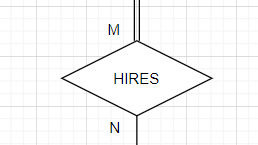
### CONTACTS (between WAITER, DELIVERY DRIVER)



Partial participation from Waiter and Delivery\_Driver. This is a binary relationship between waiter and driver where 1 waiter can contact many drivers, and a driver is contacted by 1 waiter who tells him to pick up the order and give him the location. Not all delivery drivers are contacted and not all waiters contact delivery drivers.

The relationship is needed to allow waiters to ask delivery drivers to come and give them the drop off location and know which Waiter contacted the Delivery Driver in case of a mistake.

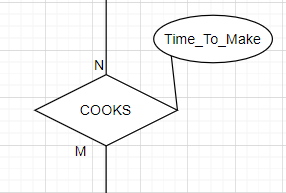
### HIRES (between HR, WAITER/DELIVERY\_DRIVER/CHEF)



Total participation from Waiter, Delivery\_Driver and Chef, partial participation from HR. This is a binary relation between HR and Waiter, HR and Delivery\_Driver and HR and Chef respectively. One HR employee can hire many chefs, waiters and drivers and one chef, waiter or driver can be hired by many HR employees as a group decision. Each chef, waiter and driver must be hired but not all HR members hire employees.

This relationship shows that each employee of the restaurant is hired by the HR department ensuring that the employees are competent enough to join the team.

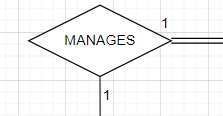
### COOKS (between CHEF, MEAL)



Partial participation from both Chef and Meal. This is a binary relationship where each Chef cooks one or more meals. A Meal is cooked by one or more chefs. A Meal is not necessarily always being cooked by a Chef (not all meals are being cooked all the time in the kitchen). Therefore, a Chef also isn’t always cooking a Meal (sometimes maybe there are no orders, or he’s at home, or at a break). The attribute Time\_To\_Make is a real time info that shows us how much time the Chef took to make the Meal.

This relationship is important to know which Chef has cooked which Meal in case of an issue with the Meal.

### MANAGES (between CHEF, KITCHEN\_STATION)



Total participation from Kitchen\_Station, partial participation from Chef. This is a binary relationship where each Kitchen\_Station must be managed by exactly one Chef. And where one Chef manages exactly one Kitchen\_Station. A Chef cannot manage multiple kitchen stations. A Kitchen\_Station can’t be managed by more than one Chef.

This relationship is important because it helps the restaurant see directly which chefs manage which kitchen stations to directly talk with the managers if needed.

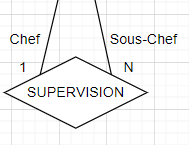
### CONTAINS (between MENU, MEAL)



Total participation from Menu and Meal. This is a binary relationship between Menu and Meal where each Menu must contain one or more meals. And each Meal must be part of one or more Menu (otherwise why is the Meal there).

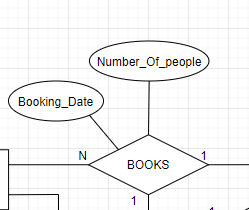
This relationship is important as it shows what are the present meals in each Menu so that the restaurant can directly use menus that contain predefined meals.

### SUPERVISION (between CHEF, SOUS-CHEF)



Partial participation from Chef from both sides. This is a binary relationship between Chef and Chef (same entity, relationship with itself). Each Chef (called then Sous-Chef) can be supervised by exactly one Chef (called Chef). Each Chef can supervise many other sous-chefs. A Chef doesn’t necessarily supervise another Chef. And a Chef isn’t necessarily supervised by another Chef.  
This relationship is important as it helps the restaurant in maintaining the hierarchy of the chefs.

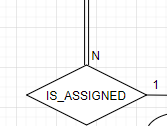
### BOOKS (between TABLE, CUSTOMER, WAITER)



Partial participation from Table, Customer and Waiter. This is a ternary relationship Table, Customer and Waiter where a Customer and a Waiter can book many Tables, a Customer books a Table via one waiter, and finally a Waiter books a Table for one Customer. The attribute Number\_Of\_people help the restaurant in keeping track of the number of people that will sit on the booked Table. The second attribute Booking\_Date helps the restaurant in the scheduling process by specifying the date and time that the Table will be booked on.

This relationship helps the restaurant in organizing its Dine-in facility

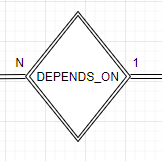
### IS\_ASSIGNED (between TABLE, WAITER)



Total participation from Table, partial participation from Waiter. This is a binary relationship between Table and Waiter, where a waiter can be assigned to many Tables, but every Table is assigned to one waiter.

This relationship helps the Restaurant in the seating process by allowing the Customer to book a Table to avoid capacity issues upon arrival.

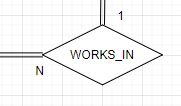
### DEPENDS\_ON (between EMERGENCY\_CONTACT, CHEF/WAITER/DELIVERY\_DRIVER)



Total participation from both Emergency\_Contact and Chef/Waiter/Delivery\_Driver. This is a binary relationship where every Emergency\_Contact depends on one Waiter/Chef/Delivery\_Driver, and every Waiter/Chef/Delivery\_Driver has many Emergency\_Contacts to depend on him.

This relationship helps the restaurant in keeping track of every employee’s emergency contact so they can contact him in case of emergency.

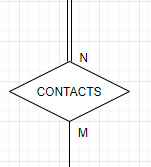
### WORKS\_IN (between CHEF, KITCHEN\_STATION)



Total participation from both Chef and Kitchen\_Station. This is a binary relationship where every Chef works in one Kitchen\_Station, and every Kitchen\_Station has many chefs working in it.

This relationship helps the restaurant by linking every Chef to a Kitchen\_Station to make work clearer and improve efficiency.

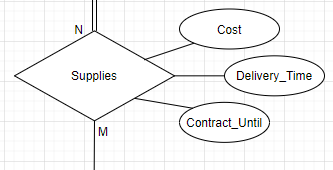
### CONTACTS (between ADMINISTRATION, SUPPLIER)



Total participation from Supplier, partial participation from Administration. This is a binary relationship where one member of the Administration contacts many suppliers, and every supplier can be contacted by many members of the administration.

This relationship facilitates the process of getting supplies for the restaurant by making a member of the administration contact a supplier.

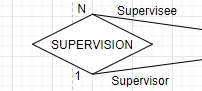
### SUPPLIES (between SUPPLIER, INGREDIENTS)



Total participation from Ingredients, partial participation from supplies. This is a binary relationship where a Supplier supplies many Ingredients, but every Ingredient can be supplied by many Suppliers. The attribute cost represents the financial cost of the supplied Ingredient by a certain Supplier. The attribute Delivery\_Time represents the time needed for a certain Supplier to deliver a certain Ingredient. The attribute Contract\_Until represents the date on which the contract made with a certain supplier ends.

This relationship helps the restaurant in keeping track of every Ingredient supplied by every Supplier alongside the Cost and Delivery\_Time to fix the financials and the inventory.

### SUPERVISION (between SUPERVISOR, SUPERVISEE)



Partial participation from both sides. This is a binary relationship between the same entity (Waiter), where a Supervisor supervises many Supervisees, but a Supervisee is supervised by one supervisor.

This relationship helps the restaurant in keeping track of the hierarchy between Waiter.

### CREATES (between CHEF, MENU)

**A diagram of a diagram

Description automatically generated with medium confidence**

Total participation from Menu, partial participation from Chef. This is a binary relationship where a Chef creates one or more Menu. Not all chefs necessarily create a Menu. But a Menu must be created by one or more chefs.

### SHIFT (between Administration, chef, waiter)

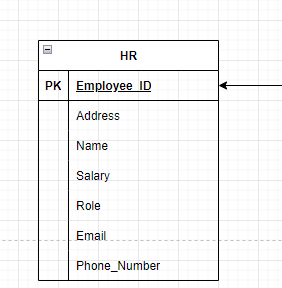


Total participation between chef and shift and waiter and shift but partial participation between shift and HR. That is, an HR member may or may not be part of the ones who set the shift times but each and every Waiter and Chef must be assigned to a shift. One HR member can assign shift(s) to many Waiters and Chefs and one chef/waiter can be assigned a shift by many HR members.

# ER to Relational Mapping

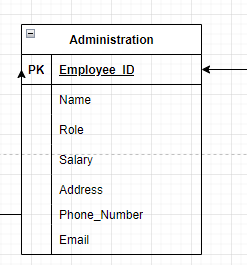
## Step 1 – Mapping Strong Entity Types

### Relation: HR



Seven columns each represent an attribute with Employee\_ID being the Primary Key. The other attributes are Address, Name, Salary, Role, Email, Phone\_Number.

### Relation: Administration



Seven columns each represent an attribute with Employee\_ID being the Primary Key. The other attributes are Address, Name, Salary, Role, Email, Phone\_Number.

### Relation: Menu

A screenshot of a menu

Description automatically generated

Six Columns each representing an attribute with Menu\_ID being the primary key. The other attributes are Image, Category, Start\_Time, End\_Time, Number\_Of\_Items.

### Relation: Meal

A screenshot of a menu

Description automatically generated

Seven columns each representing an attribute with Meal\_Name being the primary key. The other attributes are Cost, Category, Price, Recipe, Description, Cooking\_Time.

### Relation: Supplier

A screenshot of a computer

Description automatically generated

Five columns each representing an attribute with Supplier\_ID being the Primary Key. The other attributes are Address, Phone\_Number, Email, Name.

### Relation: Ingredient

A screenshot of a graph

Description automatically generated

Five columns each representing an attribute with Inventory\_ID being the primary key. The other attributes are Minimun\_Qty, Price, Stck\_Qty, Name.

### Relation: Chef

A graph with text on it

Description automatically generated

Seven columns each representing an attribute with EmployeeID being the Primary Key. The other attributes are Role, Salary, Hire\_Date, CV. We also have Works\_In which is a Foreign Key that links this relation to the relation Kitchen station. Finally, we have SupID which is a Foreign Key that links the relation Chef to itself (Recursive relationship).

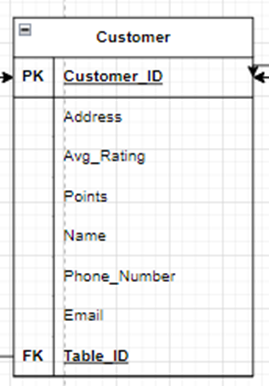
### Relation: Delivery\_Driver

A diagram of a delivery driver

Description automatically generated

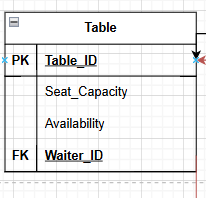
Six columns each representing an attribute with EmployeeID being the Primary Key. The other attributes are Name, Contact\_Info, Vehicule\_Type, Average\_Rating. We also have Waiter\_Contact which is a Foreign\_Key that links this relation to the relation Waiter.

### Customer



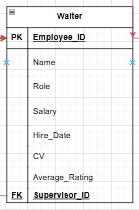
This relation has 8 columns. CustomerID (Primary Key), address, avg\_rating, points, name, phone\_number and Email. Table ID (Foreign Key) is used to link relation Customer to relation Table.

### Table



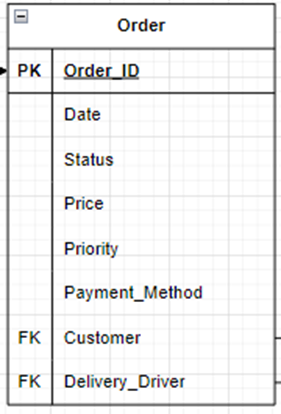
This relation has 8 columns. Table ID (Primary key), Seat capacity and Availability. Waiter ID (Foreign key) links the relation Table to the Relation Waiter.

### Waiter



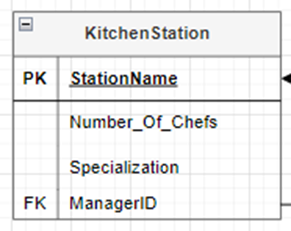
This relation has 9 columns. Employee ID (Primary Key), name, role, salary, hire date, cv, average rating. Supervisor ID (Foreign key) links the relation Waiter to itself (recursive relationship).

### Order



This relation has 8 columns. Order ID (Primary Key), date, status, price, priority, payment method. Customer ID (Foreign Key) links relation Order to relation Customer and Delivery Driver links the relation Order and the relation Delivery Driver.

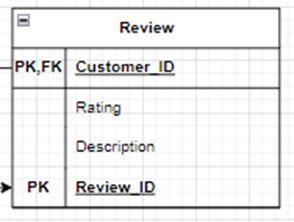
### Kitchen\_Station



This relation has 4 columns. Station Name (Primary Key), Number of chefs, specialization. Manager ID (Foreign Key) links relation Kitchen Station to relation Chef.

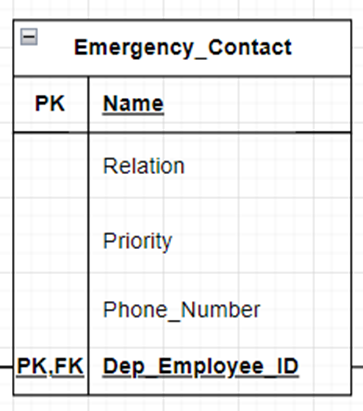
## Step 2 - Mapping of Weaking Entity Types

### Review



This relation has 4 columns. Review ID (Primary Key), Customer ID (Primary Key), rating, description. Customer ID (Foreign Key) links relation Review to relation Customer.

### Emergency\_Contact



This relation has 5 columns. Name (Primary Key), Dep\_Employee ID (Primary Key) , relation, priority, phone number. Dep\_Employee ID (Foreign Key) links relation Emergency Contact to relations Waiter, Delivery Driver and Chef.

## Step 3 – Mapping of Binary 1:1 Relationship Types

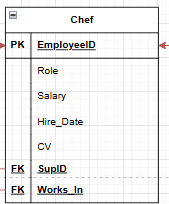
### MANAGES



The relationship MANAGES is a 1:1 relationship between chef and kitchen station. One chef manages one station and is managed by one chef. Since we have the choice between which relation to hold the FK, we chose to set the FK ManagerID in the relation KitchenStation which points to the PK EmployeeID in chef.

## Step 4 – Mapping of Binary 1:N Relationship Types

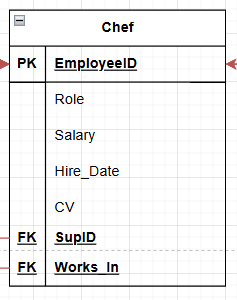
### WORKS IN



The relationship WORKS IN is a 1:N relationship between Chef and KitchenStation.

The FK Works\_In set on the N side of the relationship is in the relation Chef and points to the PK Station Name in the relation KitchenStation

### SUPERVISION



The relationship SUPERVISION is a 1:N recursive relationship. The FK SupID set in the Chef relation points to the PK EmployeeID in the relation itself.

### DEPENDS ON



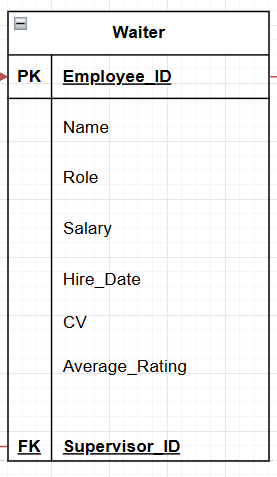
The relationship DEPENDS\_ON is a 1:N used between 4 relations: Chef, Waiter Delivery driver with Emergency\_Contact. The FK Dep\_Employee\_ID set in the N side of the relationship in the relation Emergency\_Contact and points to the PK Employee\_ID in all 3 relations Chef, Waiter and Delivery Driver.

### Contacts



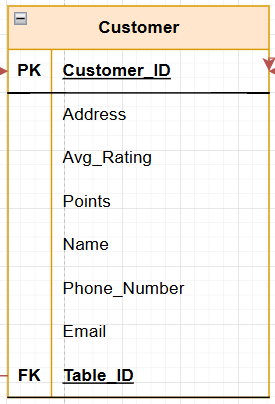
The relationship CONTACTS is a 1:N relationship between Waiter and Delivery driver. The FK Waiter\_Contact is set on the N side of the relationship in the relation Delivery\_Driver and points to the PK Employee\_ID in the relation waiter.

### SUPERVISION



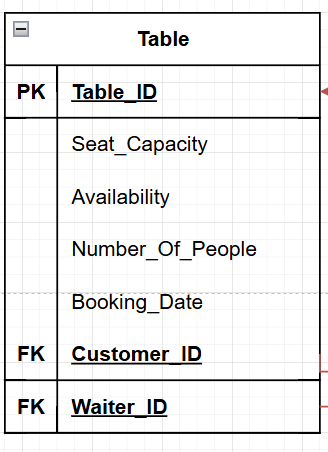
The relationship SUPERVISION is a 1:N recursive relationship. The FK Supervisor\_ID set in the Waiter relation points to the PK EmployeeID in the relation itself.

### IS\_SITTING



The relationship IS\_SITTING is a 1:N relationship between Customer and Table. The FK Table\_ID is set on the N side of the relationship in the relation Customer and points to the PK Table\_ID in the relation Table.

### IS\_ASSIGNED



The relationship IS\_ASSIGNED is a 1:N relationship between Table and Waiter. The FK Waiter\_ID set in the N side of the relationship in the Table relation points to the PK Waiter\_ID in the relation Waiter.

### WRITES



The relationship WRITES is a 1:N relationship between Review and Customer. The FK Customer\_ID is set in the N side of the relationship in the relation Review an points to the PK Customer\_ID in the relation Customer.

### PAYS



The relationship PAYS is a 1:N relationship between Customer and Order. The FK Customer\_ID is set in the N side of the relationship in the relation Order and points to the PK Customer\_ID in the relation Customer.

### DELIVERS



The relationship DELIVERS is a 1:N relationship between Order and Delivery driver. The FK Delivery\_Driver is set in the N side of the relationship in the relation Order and points to the PK Employee\_ID in the relation Delivery\_Driver.

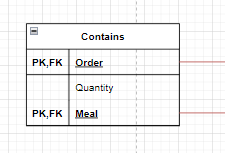
## Step 5 – Mapping of Binary M:N Relationship Types

### Contacts



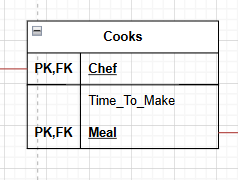
This is a relationship between Administration and Supplier. Since it is Many-to-Many, we’ve created a new relation for it with foreign keys from the Administration and Supplier relations, together they form the composite key for this relation.

### Contains



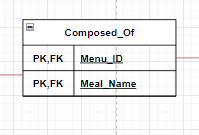
This is a relationship between Order and Meal. Since it is Many-to-Many, we’ve created a new relation for it with foreign keys from the Order and Meal relations, together they form the composite key for this relation. It also has an attribute Quantity added to it

### Cooks



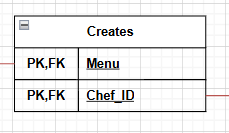
This is a relationship between Meal and Chef. Since it is Many-to-Many, we’ve created a new relation for it with foreign keys from the Chef and Meal relations, together they form the composite key for this relation.

### Composed\_Of



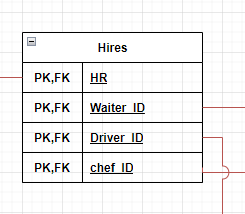
This is a relationship between Meal and Menu. Since it is Many-to-Many, we’ve created a new relation for it with foreign keys from the Menu and Meal relations, together they form the composite key for this relation.

### Creates



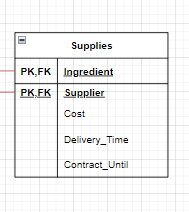
This is a relationship between Menu and Chef. Since it is Many-to-Many, we’ve created a new relation for it with foreign keys from the Menu and Chef relations, together they form the composite key for this relation.

### Hires



This is a relationship between HR and Employee\_ID. Since it is Many-to-Many, we’ve created a new relation for it with foreign keys from the HR and Employee relations (Waiter, Chef and Delivery\_Driver), together they form the composite key for this relation.

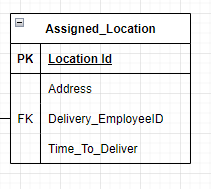
### Supplies



This is a relationship between Ingredient and Supplier. Since it is Many-to-Many, we’ve created a new relation for it with foreign keys from the Ingredient and Supplier relations, together they form the composite key for this relation. It also has three attributes added to it, which are Cost, Delivery\_Time, Contract\_Until.

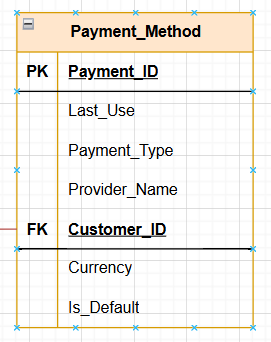
## Mapping of Multivalued Attributes

### Assigned\_Location



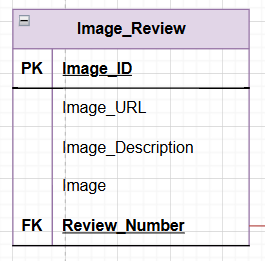
This is a Multivalued Attribute of the relation Delivery\_Driver. It has 4 columns: Location\_ID (Primary Key), Address, Time\_To\_Deliver. Delivery\_EmployeeID (Foreign Key) links the multivalued attribute to the relation Delivery\_Driver.

### Payment Method



This is a multivalued attribute of the relation Customer. It has 7 columns: Payment ID (Primary Key), last use, payment type, provider name, currency, is default. Customer ID (Foreign key) links the multivalued attribute Payment method to relation Customer.

### Image



This is a multivalued attribute of the relation Review. It has 5 columns: Image ID (Primary Key), Image URL, Image description, image. Review Number (Foreign Key) links the multivalued attribute Image to the relation Review.

### Image



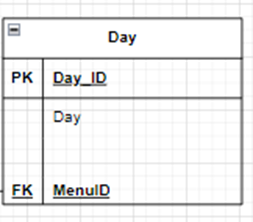
This is a multivalued attribute of the relation Meal. It has 5 columns: Image ID (Primary Key), Image URL, Image description, image. Review Number (Foreign Key) links the multivalued attribute Image to the relation Meal.

### Product



This is a multivalued attribute of the relation Product. It has 5 columns: Product ID (Primary Key), product name, product price, product shelf life. Supplier ID (Foreign Key) links the multivalued attribute Product to the relation Supplier.

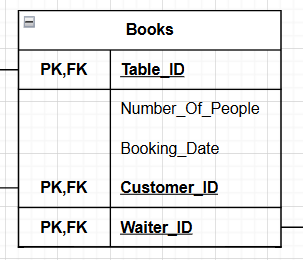
### Day



This is a multivalued attribute of the relation Menu. It has 3 columns: Day ID (Primary Key), Day. Menu ID (Foreign Key) links the multivalued attribute day to the relation Menu.

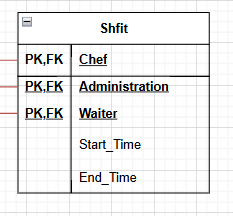
## Mapping of N-Ary Attributes

### Books



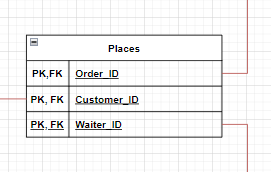
This is a relationship between Table, Waiter and Customer(ternary). Since it is Many-to-One-to-One, we’ve created a new relation for it with foreign keys from the Table, Waiter and Customer relations, together they form the composite key for this relation. It also has attributes Number\_Of\_People and Booking\_Date.

### Shift



This is a relationship between Waiter, Chef and Administration (ternary). Since it is Many-to-Many-to-Many, we’ve created a new relation for it with foreign keys from the Waiter, Chef and Administration relations, together they form the composite key for this relation. It also has attributes Start\_Time and End\_Time.

### Places



This is a relationship between Order, Customer and Waiter (ternary). Since it is Many-to-Many-to-Many, we’ve created a new relation for it with foreign keys from Waiter, Order and Customer relations, together they form the composite key for this relation.

## Step 8- Mapping Aggregation, Specialization Relationships

We do not have any special relationships from outside the course’s scope.

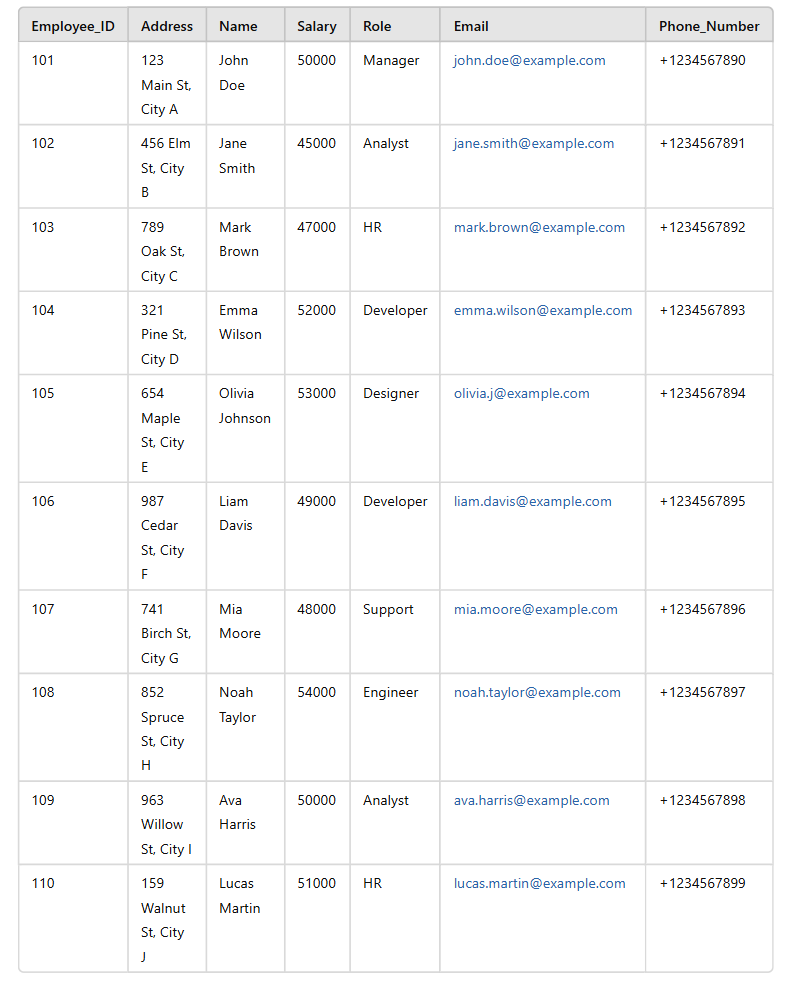
# Final Display - All Tables

A diagram of a computer

Description automatically generated

# Tables’ States

## HR



## Administration

A white sheet with many numbers and letters

Description automatically generated with medium confidence

## Menu

A screenshot of a menu

Description automatically generated

## Meal

A white table with black text

Description automatically generated

## Supplier

A screenshot of a phone number

Description automatically generated

## Ingredient

A screenshot of a table

Description automatically generated

## Chef

A table of work tasks

Description automatically generated with medium confidence

## Delivery\_Driver

A screenshot of a computer

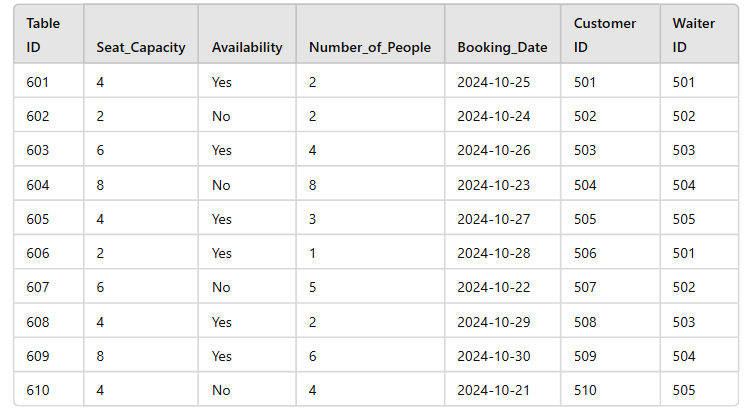
Description automatically generated

## Customer

A screenshot of a computer

Description automatically generated

## Table



## Waiter

A table of numbers and letters

Description automatically generated with medium confidence

## Order

A screenshot of a white sheet

Description automatically generated

## Kitchen\_Station

A table with a number of chefs

Description automatically generated

## Review

A screenshot of a survey

Description automatically generated

## Emergency\_Contact

A screenshot of a computer

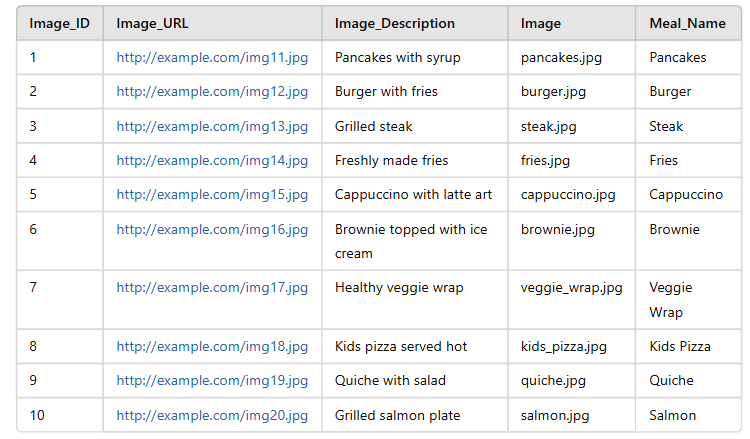
Description automatically generated

## Payment\_Method

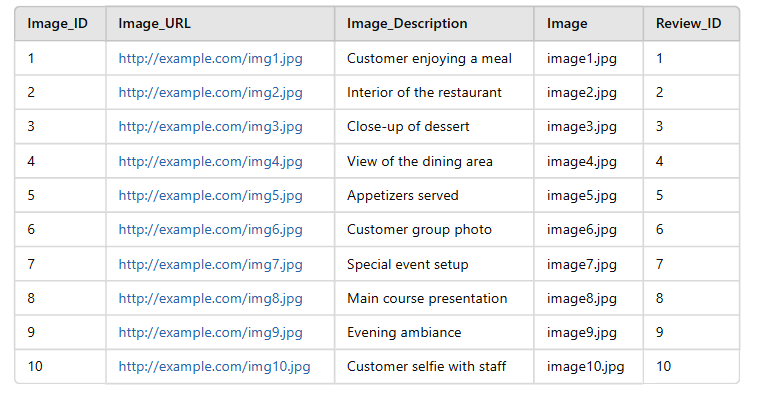
A screenshot of a computer

Description automatically generated

## Image\_Meal



## Image\_Review



## Product



## Day

A screenshot of a calendar

Description automatically generated

## Assigned\_Location

A screenshot of a computer

Description automatically generated

## Supplies

A screenshot of a white sheet

Description automatically generated

## Creates

A screenshot of a computer

Description automatically generated

## Composed\_Of

A screenshot of a menu

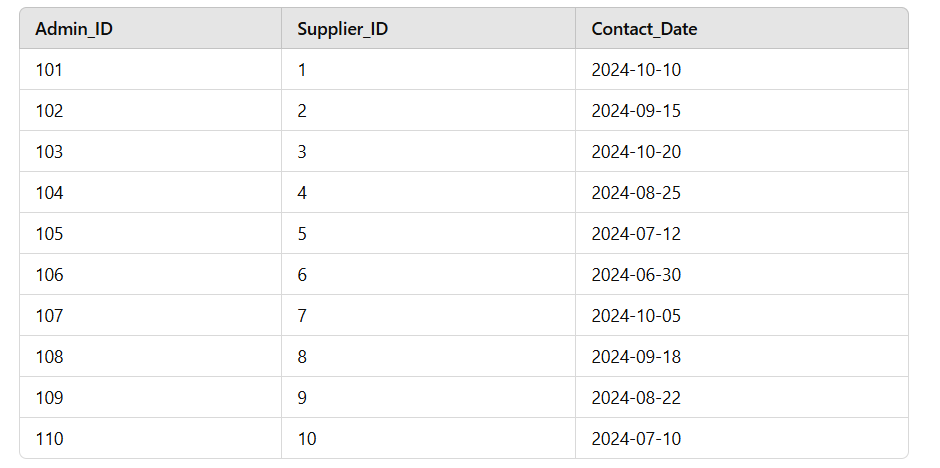
Description automatically generated

## Cooks

A table with a list of food

Description automatically generated

## Contacts



## Places

A screenshot of a data

Description automatically generated

## Hires

A screenshot of a computer

Description automatically generated

## Shift

A screenshot of a computer

Description automatically generated

## Books

A screenshot of a computer

Description automatically generated

## Contains

A screenshot of a menu

Description automatically generated

# Conclusion

As a conclusion, the proposed Entity-Relationship model for the Bakri’s Big Burger restaurant management system is designed to meet all necessary requirements to promote excellent functioning of every fraction of the restaurant. It optimizes key operational areas including but not limited to customer service, staff, management and inventory control. This system aims to enhance performance, improve workflow and facilitate decision making whether at the staff or administration level, ultimately contributing to a more efficient and well-managed restaurant environment for all stakeholders.

The next step of the design of this database is the mapping phase where the ER model will be transformed into a relational model. This process mainly involves converting the entities, attributes and relationships into tables to better structure the database and prepare it for efficient storage and querying.

# Instructor’s Feedback